



Institutional Repository - Research Portal Dépôt Institutionnel - Portail de la Recherche

researchportal.unamur.be

University of Namur

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Système de gestion de Workflow modélisation et implémentation

Colot, Pierre

Award date:
2005

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 23. Jun. 2020

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année Académique 2004-2005

Système de gestion de Workflow : Modélisation et Implémentation

Pierre Colot

Mémoire présenté en vue de l'obtention du grade de
Maître en Informatique .

Résumé

Depuis l'apparition du Business Process Reengineering vers 1985 jusqu'à aujourd'hui, l'approche processus a fait son chemin à travers les organisations et a provoqué l'apparition de nouvelles technologies de l'information. Que l'on soit tourné vers l'organisation ou vers la technologie, l'orientation processus est devenue plus qu'un avantage sur ses concurrents : une pensée primordiale, un facteur de réussite.

C'est dans ce contexte que se situe ce mémoire : il introduit le sujet en retraçant l'apparition des business processes dans les organisations, il définit les notions de business process et de workflow, il explore les concepts du domaine des workflows à travers la présentation du modèle de référence des workflows, il montre ce qu'est une définition de processus dans deux langages (BPMN, un langage graphique, et XPDL, un langage orienté XML) et propose un travail pratique de conception et de programmation d'un nouveau système d'exécution de workflow. Ce système comprend une application de gestion de listes de travaux, une application de contrôle des processus et un moteur de workflow permettant l'importation d'une définition de processus écrite en XPDL.

Mots clés : processus, workflow, système d'exécution de workflow, définition de processus

Abstract

Since the Business Process Reengineering appeared in about 1985, the process orientation has gained ground in organizations and is responsible for the emergence of new information technologies. Whether you are business-oriented or systems-oriented, process orientation has become more than an advantage over competitors : a mainstream thinking, fundamental to organizational performance.

This dissertation deals with process, workflow and associated technologies. It begins with a description of the emergence of business processes in organizations, it gives a definition of business process and workflow. It explores the specific workflow related concepts by introducing the workflow reference model. It shows how to define a process in two languages : BPMN diagrams and XPDL, a XML-oriented process definition language. It ends with a practical work : the implementation of a new workflow management system. This system includes a worklist handler application, a process control application and a workflow engine, allowing the importation of process definitions written in XPDL.

Keywords : business process, workflow, workflow management system, process definition

Je tiens à remercier particulièrement mon promoteur Jean-Marie Jacquet pour ses encouragements, sa patience, ses conseils avisés, et la relecture critique de mon mémoire.

Je remercie également mes parents et amis pour leur soutien et les relectures de ce mémoire.

Table des matières

1	Introduction : Business Process et Workflow	1
1.1	De l'approche fonctionnelle ...	2
1.2	... Vers l'approche processus ...	4
1.2.1	1985 - 1993 : l'apparition du Business Process Reengineering ...	4
1.2.2	1994 - 1995 : Le déclin du Business Process Reengineering ...	4
1.2.3	Une vision plus équilibrée ...	4
1.2.4	1996 - aujourd'hui : Après le Business Process Reengineering ...	6
1.2.5	BPR et Kaizen ...	6
1.3	L'adaptation des organisations à cette évolution ...	6
1.3.1	La maturité des méthodes d'analyse et de modélisation de processus ...	7
1.3.2	La maturité des systèmes de gestion de workflow ...	9
1.3.3	Conclusion ...	9
2	Les termes de base du monde des workflows	11
2.1	Business Process ou Workflow? ...	13
2.2	Les définitions du dictionnaire ...	13
2.3	Les définitions extraites du jargon informatique ...	14
2.4	Les termes utilisés lors de la conception d'un processus ...	15
2.4.1	Pourquoi ont-ils revus leurs termes? ...	15
2.4.2	La définition des (re-) concepteurs de business process ...	16
2.5	Les définitions de la WfMC ...	17
2.5.1	Que sont les Workflow Management Systems? ...	18
2.5.2	La terminologie du WfMC ...	18
2.5.3	Qu'est ce qu'un workflow? ...	19
2.5.4	Pourquoi utiliser des Workflow Management Systems? ...	21
2.5.5	Les différentes catégories de fonctions d'un WFMS ...	22
2.5.6	Les interfaces d'un WFMS ...	26

2.6	Conclusion	26
3	Le Modèle de référence des Systèmes de Gestion de Workflow	29
3.1	Introduction	30
3.2	Structure générique d'un WFMS	31
3.3	Le Workflow Reference Model	31
3.3.1	Qu'est ce qu'un service d'exécution de workflow?	32
3.3.2	Qu'est ce qu'un moteur de workflow?	34
3.3.3	Les transitions d'état des instances de processus et d'activités	34
3.3.4	Le routage des activités	38
3.3.5	L'enchaînement de deux instances d'activités	40
3.3.6	Les données dans le WFMS	42
3.3.7	L'échange de données (direct - indirect)	44
3.4	La WAPI (Workflow Application Programming Interface)	45
3.4.1	Interface 1 : l'importation et l'exportation de définition de processus	45
3.4.2	Interface 2 : l'interface avec les applications clientes	48
3.4.3	Interface 3 : l'interface avec les applications invoquées	53
3.4.4	Interface 4 : l'interopérabilité entre systèmes de workflows	55
3.4.5	Les API d'interopérabilité de services d'exécution de workflow	58
3.4.6	Interface 5 : L'administration système	61
3.5	Conclusion	62
4	BPMN et XPD	63
4.1	Vers une définition visuelle d'un processus	63
4.2	BPMN : Une notation standard pour les processus d'une organisation	64
4.2.1	Les objets de flux	65
4.2.2	Les objets de connexion	66
4.2.3	Les couloirs ("swimlanes")	67
4.2.4	Les artefacts	68
4.3	XPD : un langage XML de description de processus	69
4.3.1	Les types abstraits ou complexes	70
4.3.2	Les acteurs participants	71
4.3.3	Les paramètres des applications	72
4.3.4	Les activités	73
4.4	La correspondance entre BPMN et XPD	73
4.4.1	Quelques patterns communs	74
4.5	Conclusion	78

5	La construction d'un système de gestion de workflow	79
5.1	La définition du processus	80
5.1.1	Les participants	81
5.1.2	Les données relatives au workflow	81
5.1.3	Les activités	83
5.1.4	Les applications	83
5.1.5	Les données des applications	86
5.2	Les composants du système	86
5.2.1	Les noeuds et les composants	87
5.3	Les interactions entre l'utilisateur et les applications front-end	88
5.3.1	Les cas d'utilisation	88
5.3.2	L'identification des acteurs UML	89
5.3.3	L'identification des cas d'utilisation	89
5.3.4	Structuration en packages de cas d'utilisation	91
5.4	Modélisation de la navigation	91
5.5	L'analyse du domaine	96
5.5.1	L'identification des objets du domaine	96
5.5.2	La structuration en package de classes	99
5.6	Les classes de conception et d'implémentation	100
5.6.1	L'application de gestion de listes de travaux	100
5.6.2	L'application de contrôle de processus	101
5.6.3	Le moteur de workflow	103
5.7	L'implémentation des composants et des données	105
5.7.1	Le moteur d'exécution de workflow	105
5.7.2	Les données relatives au workflow	107
5.7.3	Les applications front-end	107
5.8	Conclusion	108
6	Conclusion	109
	Annexes	111
A	Définition de processus en XPDL	111
	Bibliographie	119

Chapitre 1

Introduction : Business Process et Workflow

Ce chapitre introductif sur les business processes (processus) et les workflows répond aux questions suivantes :

1. Qu'est ce qu'un Business Process ?
2. Quel est le lien, dans le cadre des organisations, entre un workflow et un business process ?
3. Comment des organisations (dont les maîtres-mots étaient, depuis un siècle, la division du travail, la spécialisation, les fonctions, et la qualification des employés) en sont-elles venues à faire machine arrière, à repenser l'organisation du travail (non pas à l'échelle d'un département ou d'une fonction, mais bien à l'échelle de l'organisation entière) suite à l'apparition d'une nouvelle idée : le Business Process Reengineering ?
4. Comment a évolué le Business Process Reengineering jusqu'à nos jours, quel est le degré de maturité de cette évolution ?

Ce chapitre a pour but d'introduire le lecteur dans le monde des workflows en exposant ses sources, c'est-à-dire l'évolution des organisations (d'une approche fonctionnelle vers une approche processus), et en exposant ce que les technologies de l'information ont développé durant cette évolution : les systèmes de gestion de workflows.

La section 1 (De l'approche fonctionnelle...) reprend les avantages et les inconvénients des organisations orientées "fonctions", et explique en quoi un changement était à envisager.

La section 2 (... vers l'approche processus) annonce l'évolution des organisations par de nouvelles idées. Elle explique les grandes lignes de ce passage difficile et en quoi les nouvelles solutions répondent aux problèmes de l'approche fonctionnelle.

La section 3 (L'adaptation des organisations à cette évolution) donne une indication sur la maturité de la nouvelle approche de l'organisation du travail. Le chapitre se termine par une brève conclusion.

1.1 De l'approche fonctionnelle ...

Les organisations fonctionnelles [9], ou orientées fonction, ont pris naissance à la révolution du début du siècle, avec l'avènement de la nouvelle théorie de l'organisation scientifique du travail, par Frederick Taylor et d'autres. Cette théorie était basée sur la division du travail en tâches simples et répétitives, et une supervision étroite des employés.

Le succès de la division du travail en spécialités a entraîné un nombre toujours grandissant de spécialistes, non seulement en manufacture mais aussi dans des domaines comme la comptabilité et le marketing. Les organisations appartenant à d'autres domaines que la manufacture, comme les banques, les compagnies d'assurances, les hôpitaux, les entreprises gouvernementales, prirent également ce tournant. Les produits plus sophistiqués ont ouvert de nouvelles perspectives de travail comme la recherche et le développement, le managing pour planifier, organiser et contrôler le travail. Ces nouvelles perspectives de travail ont rapidement eu la réputation de n'être là que pour augmenter l'efficacité des vrais travailleurs responsables de la production, mais ils ont en fait permis à toute une génération d'organisations d'exister et de prospérer.

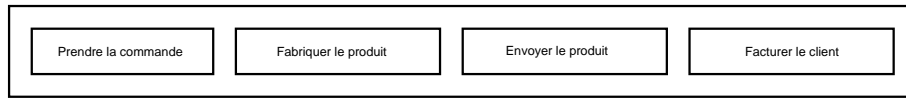
Comment les managers arrivaient-ils à contrôler tous ces travaux? Le personnel était divisé en groupe de spécialistes ou "fonctions" comme la comptabilité, l'ingénierie, la manufacture. Ne se préoccupant pas du reste de l'organisation, les groupes isolés offraient peu de résistance au travail des managers (qui était de gérer les ressources, planifier les tâches, contrôler le travail), et les organisations "orientées fonction" ont pu dominer le siècle passé.

Qu'est-ce au juste qu'une "fonction" de l'organisation? L'organisation fonctionnelle est tellement implantée dans les esprits que beaucoup penseront que fonction et organisation sont des synonymes, donc que les fonctions d'une organisation sont les différents départements qui composent l'entreprise. Avec une nuance toutefois : un département est une structure rassemblant un *groupe de personnes* et d'autres ressources pour atteindre un but commun. Une fonction, par contre, est un domaine regroupant les *travaux d'une même nature* et requérant un savoir et des aptitudes spécifiques. Ces définitions mettent en lumière le problème central des organisations basées sur des spécialités fonctionnelles isolées dans des départements : en mettant tant l'accent sur l'efficacité de la division du travail, quelle attention était portée au but, au résultat final, que ces fonctions et département, divisés et isolés, cherchaient à produire?

Les inconvénients des organisations fonctionnelles

En général, dans une organisation, les employés (organisés en départements) font de leur mieux pour contribuer à la réussite de leur entreprise, et donc ils optimisent le travail (les fonctions) dont ils sont responsables. Mais optimiser des parties isolées n'optimise pas forcément le tout. Les employés ne voient pas l'organisation du travail dans l'entièreté de l'entreprise, donc peuvent par inadvertance porter préjudice au résultat final en essayant d'optimiser leur travail.

Prenons un exemple : voici un processus qui remplit une commande pour un produit (figure 1.1). Le processus est composé de quatre étapes : prendre la commande, fabriquer le produit, envoyer la commande, facturer le client.

FIG. 1.1 – *Les étapes d'un processus de traitement d'une commande.*

L'étape "prendre la commande" devrait rassembler toutes les informations pour fabriquer, envoyer et facturer le produit, même si cela ralentit la prise de la commande. La fabrication devrait être menée à un niveau de détail suffisant pour pouvoir facturer les coûts de production au client et analyser le processus de fabrication en vue de l'améliorer, mais cela cause aussi des délais. Le service d'envoi et de facturation préféreraient traiter tout un groupe d'opérations d'un coup car ce serait plus efficace, mais cela conviendrait-il au client et à l'entreprise? Chaque partie a une perspective limitée du processus et améliorer une part pourrait écarter la possibilité d'améliorer le service entier.

Autrement dit, chaque groupe de personnes travaillant pour une fonction de l'entreprise peut définir son travail ou celui de son département comme étant un processus. Bien sûr, chacun de ces travaux est en fait une fonction intervenant dans une partie ou l'autre d'un processus de production "complet". On peut facilement deviner ce qui se passe si chaque personne optimise son mini-processus de travail. Dans les limites de ce mini-processus, le travail sera efficient du point de vue de celui qui le fait, en grande partie parce que les inputs et outputs de cette fonction sont optimales pour l'acteur responsable du travail, mais gênent le client (celui qui sollicite cette fonction). L'effort d'adaptation imposé au client qui en résulte nuit au processus réel et le processus complet s'exécutera moins bien qu'avant. En conclusion : une amélioration dans un département particulier provoque un goulot d'étranglement dans un autre.

Pour résumer, la spécialisation a entraîné des gains énormes en efficience, mais a apporté son lot de problèmes. Ceux-ci peuvent être identifiés dans une de ces 3 catégories :

1. le processus complet est tellement éclaté qu'il n'est plus visible, donc il n'est plus mesurable ni améliorable. Il est toujours là, mais il est caché par une trop forte focalisation sur l'aspect fonctionnel et sur les spécialités.
2. Les améliorations données à chacune des activités ont bénéficié à une fonction au détriment du processus. Il était supposé que si les départements travaillaient bien individuellement, il en sera de même pour l'organisation entière. En fait, tout travail améliorant seulement une partie de l'organisation entraîne un goulot d'étranglement ailleurs ou n'apporte pas de valeur au client ou à l'entreprise.
3. Quand un objet passe entre les étapes d'un processus éclaté, le passage de main en main (les "handoffs") entre les groupes de spécialistes causent des délais, des erreurs et des dépenses. Pour qu'une fonction puisse traiter une demande aussi efficacement que possible, une grande quantité de demandes est rassemblée pour être traitée d'une certaine manière avant d'entrer dans l'unité fonctionnelle, ce qui cause un impact négatif sur le processus complet.

C'est pour réagir à ces problèmes qu'est né le BPR (Business Process Reengineering).

1.2 ... Vers l'approche processus

1.2.1 1985 - 1993 : l'apparition du Business Process Reengineering

L'idée du Business Process Reengineering était d'identifier, de rendre visible, de repenser et reconcevoir les Business Process entiers pour les améliorer. L'organisation serait vue comme un tout et les inefficiences et les irrationalités seraient éliminées. Les mesures prises sur les tâches individuelles, comme sur la vérification d'un formulaire ou le nombre d'appels téléphoniques gérés par heure, ont fait place à un accent mis sur une valeur, un but à atteindre, comme des livraisons de produits faites dans les temps, ou la satisfaction du consommateur. Un facteur crucial était l'utilisation de technologies avancées.

Les experts ont identifié la raison de l'échec : l'utilisation inadéquate de ces technologies. Les organisations gardaient leurs processus intacts et n'utilisaient la technologie que pour essayer de les accélérer. Les business processes devaient être repensés avec une vision "au travers" des fonctions de l'organisation. Les technologies ne doivent pas servir à automatiser les processus existants mais à permettre à de nouveaux processus, adaptés à ces technologies, d'exister.

Le Reengineering a été critiqué pour avoir prétendu que détruire la sur-spécialisation induite par la révolution industrielle pour faire place à des tâches réunifiées et un processus visible était nécessairement une bonne chose. La révolution industrielle n'avait-elle pas apporté d'incroyables bénéfices aux organisations? Les organisations orientées fonction avaient été critiquées et devaient évoluer, mais n'avaient-elles pas procuré beaucoup d'avantages pour les consommateurs? Était-ce vrai qu'il y avait dans les organisations des employés travailleurs mais n'étant que des poids morts? Reconcevoir les processus complets en utilisant des sous-processus fournis par des sous-traitants allait-il vraiment générer une diminution des coûts? Ces questions ont été remises à l'honneur quand les problèmes de l'approche processus ont émergé.

1.2.2 1994 - 1995 : Le déclin du Business Process Reengineering

Il s'est avéré que reconcevoir les processus était faisable mais les implémenter était difficile. 70 à 80% des projets de BPR ont échoué. Le reengineering est passé pour une application insensée des technologies informatiques, n'entraînant que la diminution du personnel et de l'outsourcing. Il est devenu très impopulaire, et le terme "reengineering" n'est plus utilisé aussi fréquemment, sauf en termes non élogieux.

Malgré cela, les entreprises consacrent du temps à la reconception de leurs processus et le taux de succès est toujours en hausse. L'orientation processus est devenu un standard dans le monde des organisations, alors que précédemment une adaptation réussie était un avantage sur les concurrents. Les "best practices" et une vision plus équilibrée de l'approche processus est apparue.

1.2.3 Une vision plus équilibrée

Pendant les premières années du BPR, il y eut beaucoup de bruit sur les faiblesses de l'organisation fonctionnelle et les forces de l'orientation processus. Après quelques

années et du recul, il est clair que toute organisation, en particulier l'organisation fonctionnelle, a ses avantages (sinon comment aurait-elle été en application si longtemps?) et que l'adaptation vers une organisation orientée processus pose des difficultés.

Pour résumer, voici une liste [9] des points positifs et négatifs d'une organisation fonctionnelle. Les avantages :

1. de plus grandes quantités produites et des économies d'échelle tout en gardant un très haut niveau de qualité,
2. les managers peuvent facilement diriger les employés,
3. la spécialisation entraîne un haut niveau de qualification,
4. la spécialités sont des domaines reconnus et les institutions peuvent former les employés jusqu'à un certain niveau leur garantissant une carrière dans ce domaine.

Les inconvénients :

1. l'attention est portée sur les tâches et sur une efficacité locale, au détriment d'une attention portée sur le processus et le résultat final,
2. le client demandant un produit ainsi que le produit final sont "invisibles" pour les départements autres que le front-office; ce manque d'attention vis-à-vis du client pourrait mener à une diminution de la qualité du service et de la satisfaction du client,
3. la communication inter-fonctionnelle n'est pas facilitée voire inexistante, ce qui peut entraîner des conflits ne pouvant être résolus que par l'intervention d'une autorité supérieure,
4. un changement requérant une coopération inter-fonctionnelle est difficile,
5. les emplois sont moins valorisants, la participation des employés est insignifiante aux yeux de ceux-ci tant les tâches sont nombreuses. Le tout nécessite un management important, une bureaucratie pour planifier, diriger et contrôler.

Le Business Process Reengineering apporte des avantages qui répondent aux inconvénients de l'organisation fonctionnelle [9], mais apporte également de nouveaux inconvénients. Les avantages :

1. une attention appropriée est donnée au client et au produit délivré,
2. un processus défini qu'on peut répéter est mesurable et améliorable,
3. le flux rationalisé et l'élimination du travail non nécessaire ou contre productif rend le processus plus efficace,
4. il n'est plus nécessaire d'avoir autant de bureaucratie (l'organisation devient horizontale plutôt que verticale) et une plus grande partie du travail fourni répond donc aux besoins du client,
5. le travail est moins divisé et les employés ont conscience de leur participation par rapport au résultat final.

Les inconvénients :

1. il est plus difficile de diriger un personnel comprenant le processus,
2. les travaux sont plus extensifs et plus exigeants, donc :
 - (a) les niveaux d'entrée définis pour les spécialistes ne s'appliquent plus, car on a besoin de personnel multi-qualifié,

- (b) la formation continue et le recyclage du personnel entraîne un plus grand niveau de stress et des pénuries de personnel qualifié,
- 3. la mise en oeuvre est difficile,
- 4. l'amélioration des processus entraîne des changements continus et un manque d'efficacité,
- 5. pour reconcevoir et remplacer les systèmes informatiques d'une organisation fonctionnelle, il faut de gros investissements.

1.2.4 1996 - aujourd'hui : Après le Business Process Reengineering

L'orientation processus s'est maintenant installée. Les organisations sont conscientes que le Business Process Reengineering est fondamental pour atteindre des performances. Plutôt que de proclamer qu'elles font du BPR, les organisations considèrent l'amélioration des processus comme étant un comportement normal.

Le Reengineering s'est donc désloganisé, on parle maintenant d'orientation processus ou de management du processus. Cette nouvelle notion est la fusion de deux tendances : le Business Process Reengineering et le Continuous Process Improvement (CPI), également désigné par au japon sous le nom de "Kaizen".

1.2.5 BPR et Kaizen

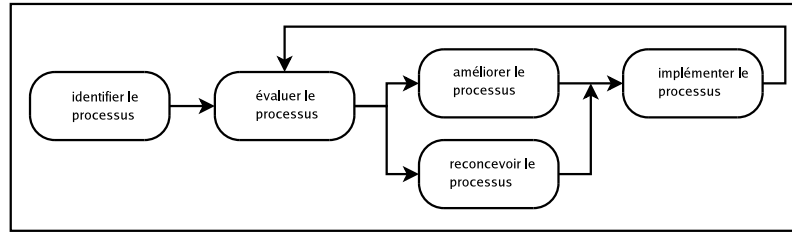
En japonais le mot "Kaizen" signifie amélioration, amélioration sans gros moyens, en impliquant tous les acteurs, des directeurs aux simples employés, et en utilisant surtout le bon sens commun. La démarche japonaise repose sur des petites améliorations faites jour après jour, mais constamment, c'est une démarche graduelle et douce qui s'oppose au concept plus "occidental" de réforme brutale du type "on jette tout et on recommence avec du neuf".

Kaizen ne se justifie pas parce que l'on peut toujours faire mieux, mais parce que le "mieux" change avec le temps. Par analogie, avant la crise pétrolière, les fabricants de voitures construisaient de grosses voitures qui consommaient 30 litres d'essence aux 100 km. Après la crise, les fabricants ont commencé à construire de plus petites voitures qui consommaient moins, le mot d'ordre était passé de "plus gros = mieux" à "plus petit = mieux".

Les partisans du BPR se demandaient pourquoi la communauté Continuous Process Improvement essayait d'améliorer des processus qui auraient dû, selon eux, être totalement remplacés. De leur côté, les Kaizeniens trouvaient l'attitude BPR (tout jeter et recommencer) assez brutale. Ces deux tendances ont été désloganisées et remplacées par le process management ou l'orientation processus (figure 1.2) : si la conception (ou la reconception) d'un processus ne se fait qu'une fois de temps en temps, l'amélioration de ce processus est continue.

1.3 L'adaptation des organisations à cette évolution

L'étude de l'adaptation des organisations à l'évolution de l'approche fonctionnelle vers l'approche processus est un sujet qui mériterait un mémoire à lui tout seul. Il

FIG. 1.2 – *L'orientation processus, née de la fusion entre BPR et Kaizen.*

est donc difficile, alors que ceci n'est qu'une introduction au monde des processus et des workflows, de couvrir ce sujet. Cette section ne présentera que deux indications (une par sous-section) sur le niveau de maturité de l'orientation processus dans une organisation. La première sous-section se rattache aux méthodes de modélisation des processus dans les organisations. L'autre sous-section provient d'un article [6] écrit par un concepteur de système de gestion de processus. Ces deux indicateurs mettent en lumière deux courants, deux métiers complémentaires : d'une part, l'analyse (et la modélisation) des processus et d'autre part, la conception (et l'implémentation) systèmes de gestion de workflow, et donnent une idée sur la maturité de ces disciplines.

1.3.1 La maturité des méthodes d'analyse et de modélisation de processus

Dans sa thèse [10], Fosdick a observé que, quand une nouvelle technologie apparaît, elle reçoit une publicité démesurée dans le monde des technologies de l'information. La courbe de publicité sur le graphique de la figure 1.3 reflète la mesure de cette publicité : le nombre d'articles sur cette technologie, le nombre d'analystes intéressés par le sujet et qui en discutent, le nombre de conversations comprenant les termes spécifiques liée à cette nouveauté, etc...

Le degré de publicité est démesuré par rapport à l'utilisation effective de la nouveauté, du moins à son apparition. Tout le monde en parle, l'adopte, mais pas à bon escient. A mesure que la nouveauté est de plus en plus utilisée, comme le décrit la courbe d'utilisation, elle fait de moins en moins la une, comme le montre la courbe de publicité en phase décroissante. Cela se produit car, bien sûr, ce n'est plus une nouveauté. Pour reprendre l'exemple de la technologie, celle-ci est à ce moment bien en place et la publicité se tourne vers la nouveauté technologique suivante.

Fosdick a expliqué cela en utilisant des exemples comme les bases de données relationnelles mais cela s'applique aussi pour les Business Process. Selon [10], dont le sujet est l'orientation processus et les business processes, nous serions dans la phase de maturité de l'orientation processus.

La première référence aux Business Process inter-fonctionnels date de 1985; vers 1990 tout le monde en entendait parler, c'était la phase de découverte.

Les premiers articles critiquant les organisation orientées fonction et encourageant une orientation processus sont apparus. Malheureusement, il ne s'agissait que divers exposés sur les concepts et la terminologie. Il s'agissait d'anciennes méthodes remises

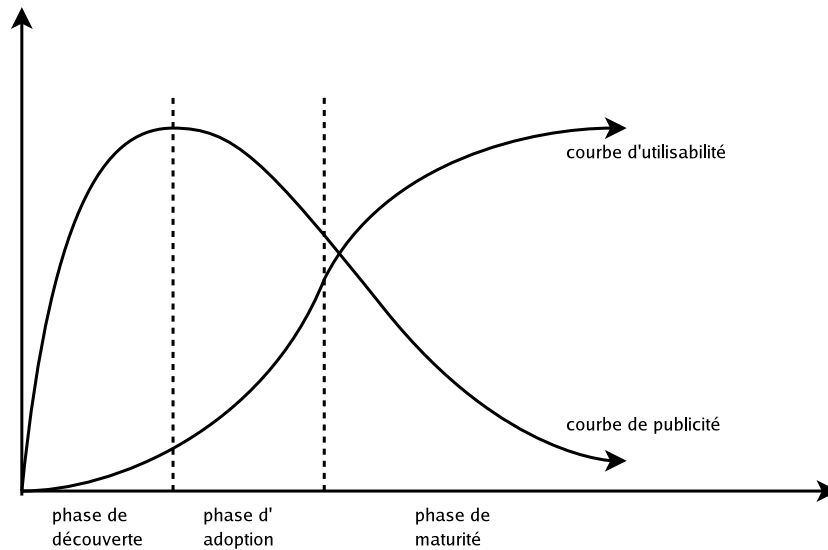
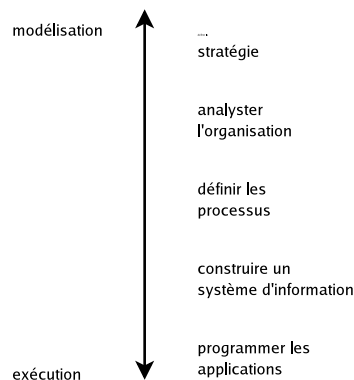


FIG. 1.3 – Les courbes de publicité et d'utilisabilité du théorème de Fosdick.

au goût du jour avec la nouvelle terminologie. Pour les différents publics, les exposés et les approches étaient différents. Par exemple, les responsables de la stratégie d'une entreprise ou les analystes de Business Process ne verront pas l'approche processus de la même manière qu'un concepteur d'architecture d'un système d'information ou un programmeur. Nous étions là dans la phase d'adoption.

En fin de compte, chacun, selon son rôle dans l'organisation (définir la stratégie, analyser l'organisation, définir les processus, ... voir figure 1.4), a sa propre vision de l'approche, ses propres termes et ses propres concepts. Les concepts de base sont les mêmes pour tout le monde, mais les concepteurs de process sont plus tournés vers des langages sous forme de diagrammes et des approches pratiques pour améliorer les processus existants. De leur côté, les informaticiens implémentent les moteurs d'exécution de ces Business Process et sont intéressés par des langages de description de processus pouvant être analysés par ces moteurs, et par de nouveaux systèmes d'information plus adaptés à l'approche processus. Chacun de ces groupes a ses propres idées, ses propres priorités.

Dans sa thèse, Fosdick justifie qu'il y a eu passage de la phase d'adoption vers la phase de maturité par le fait que l'approche processus a été adoptée avec succès dans un nombre croissant d'organisations et qu'elle ne fait plus la une des journaux en tant que "nouveau". De leur côté, les praticiens de l'approche processus, les hommes de métier, se mettent à élaborer des guides contenant des informations pratiques reconnues sur le sujet, des best practices, pour faire face aux problèmes récurrents qu'ils rencontrent car des techniques efficaces et adéquates, mises au point après la phase de découverte et la phase d'adoption, ne sont pas suffisamment répandues, et que la plupart des articles sont consacrés aux nouvelles idées.

FIG. 1.4 – *Les acteurs impliqués dans le monde des business processes.*

1.3.2 La maturité des systèmes de gestion de workflow

Du côté de l'ingénierie du logiciel et des systèmes d'information, la notion de business process est intimement liée au concept de workflow, qui est une vision automatisée et informatisée d'un business process. Le "Workflow Management" suggère l'idée d'une série d'interactions entre des utilisateurs et un système de gestion de processus (ou de workflow), il est utilisé dans un contexte technique. Le "Business Process Management" ne se limite pas à cela et englobe l'analyse du processus actuel, l'impact organisationnel d'un changement dans ce processus, le point de vue d'un manager.

Tom Baeyens [6] considère que les workflows (et donc la gestion informatique, l'automatisation des business process) ne sont qu'à la phase initiale de maturité. Il justifie cela par le fait que si l'on parle de bases de données relationnelles, tout le monde dans le métier voit clairement de quoi il s'agit. Quand on parle de workflow, ces mêmes personnes de métier verront de quoi il s'agit, mais tout le monde aura un avis, une vision et une terminologie différents. Il illustre cette comparaison avec la courbe des attentes de la technologie (technology hype curve, figure 1.5).

Une observation significative est le nombre disproportionné de concepts utilisés. Aucun outil ou aucune spécification de ce domaine n'est le (la) même. On tend à recouvrir tous ces concepts d'une couche supérieure servant à traduire ces concepts d'un outil à un autre outil, ou d'une spécification à une autre spécification. Il n'est pas rare qu'un terme utilisé chez l'un ait une autre signification chez l'autre.

1.3.3 Conclusion

L'orientation processus est devenue aujourd'hui une pensée primordiale pour les organisations. L'amélioration des Business Processes et l'amélioration des technologies de l'information sont inséparables, car un projet de reconception de processus implique toujours un effort d'adaptation au niveau des systèmes d'information. Et cela est vrai également dans l'autre sens : l'implémentation d'un nouveau système d'information conséquent implique presque toujours une reconception des processus.

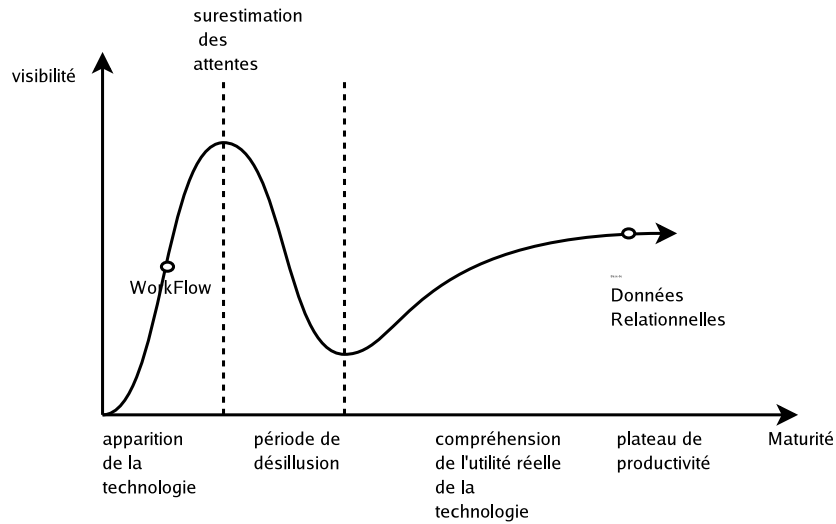


FIG. 1.5 – La courbe des attentes de la technologie (technology hype curve).

Les workflows sont apparus en même temps que les business processes et sont intimement liés à ceux-ci. Ils représentent la base sur laquelle reposent les nouveaux systèmes d'information basés sur les business processes que sont les systèmes de gestion (ou d'exécution) de workflow. Cependant, alors que les business processes atteignent leur maturité, les workflows sont encore en phase de développement. Il y a donc un décalage entre les attentes des clients de cette nouvelle technologie et ce qu'elle peut apporter à ce jour.

Ce chapitre s'est inspiré des premiers chapitres du livre *Workflow Modeling Tools for Process Improvement and Application Development* d'Alec Sharp et Patrick McDermott [1] pour l'histoire du processus, du Business Process Reengineering et de Kaizen, et de la thèse de Fosdick.

Chapitre 2

Les termes de base du monde des workflows

On ne parle pas du business process et du workflow de la même manière dans les différents métiers qui sont concernés. Chaque métier a son propre dictionnaire et ses propres outils de travail, liés aux activités qui leur sont imparties : certains établissent la stratégie et définissent les points saillants à améliorer dans les processus de production pour rendre un meilleur service au client, d'autres analysent l'organisation, d'autres dessinent des BPD (business process diagrams), d'autres conçoivent un système d'information adapté aux changements constants dans les processus, d'autres parlent implémentation du système en utilisant au mieux les technologies informatiques actuelles.

Les termes utilisés dans les différents métiers concernés par les business process et les workflows ont parfois plusieurs significations (autrement dit aucune signification) selon les métiers, les organisations et même les différents systèmes de gestions de workflow. Ce second chapitre montre les divergences au niveau des termes entre les métiers et reprend le début d'une liste de définitions (standardisée par un organisme compétent) pour les termes de base utilisés dans le monde des workflows.

La section 1 "Business Process ou Workflow?" montre ce en quoi ces deux termes inséparables sont différents.

La section 2 "Les définitions du dictionnaire" reprend les définitions d'un dictionnaire (anglais) et montre en quoi ces définitions ne conviennent pas.

La section 3 "Les définitions extraites du jargon informatique" tente d'obtenir de meilleures définitions en puisant dans le glossaire du domaine de l'informatique, espérant y trouver plus de rigueur. Les définitions obtenues ne sont malgré tout encore insatisfaisantes.

La section 4 "Les termes utilisés lors de la conception d'un processus" reprend cette fois les termes utilisés par les analystes qui améliorent et reconçoivent les processus selon de nouveaux buts, de nouvelles stratégies.

La section 5 "Les définitions du WfMC" présente la Workflow Management Coalition (WfMC) qui a mis au point les premières définitions d'une liste qui sera prise comme référence tout au long de ce mémoire, ainsi qu'une structure générique pour

les systèmes de gestion de workflow.

2.1 Business Process ou Workflow?

Un sujet qui mérite une attention particulière est la relation entre deux concepts inséparables de nos jours : un processus d'une organisation (business process) et un workflow. Ils ont une base commune : ils se rapportent à l'organisation d'une succession de travaux effectuées par des acteurs d'une organisation, mais leur signification précise diffère selon les différentes disciplines, les différents métiers. Ceci soulève un premier problème : un même terme n'a pas la même signification selon qu'on appartient à l'un ou l'autre métier.

On parle plus souvent de "business process modeling" ou "workflow modeling" dans le milieu des analystes des organisations et des (re-)concepteurs de processus. Quand on parle de processus, on observe le travail à travers l'ensemble de l'organisation, en passant les frontières des départements, depuis le moment où le client demande un service jusqu'au moment où le client est facturé. Le Business Process Management reprend tout d'abord l'analyse de l'organisation elle-même. Il commence par l'analyse et l'évaluation de la manière dont elle organise à ce jour ses services (le processus "actuel"), les impacts d'une modification dans cette organisation, la transition vers une nouvelle version du processus et du système d'information sous-jacent. Pour ces analystes, le workflow n'est que le chemin que tracent les tâches d'une extrémité à l'autre du processus.

On parle de "gestion du workflow" dans le milieu des technologies de l'information. Ce terme dénote les aspects informatiques, techniques, et non pas organisationnels comme le processus. Plus particulièrement, ce terme dénote une série d'interactions entre les acteurs de l'organisation et le système informatique. On peut considérer le workflow comme étant l'exécution d'un processus à l'aide d'un système d'information approprié.

2.2 Les définitions du dictionnaire

Pour établir une liste comprenant les définitions des termes liés au domaine des workflow, commençons par regarder au dictionnaire [2] si ces termes existent, et comment ils sont reliés entre eux.

Process

- a natural phenomenon marked by gradual changes that lead toward a particular result
- a natural continuing activity or function
- a series of actions or operations conducing to an end; a continuous operation or treatment especially in manufacture

Function

- professional or official position : occupation
- the action for which a person or thing is specially fitted or used or for which a thing exists : purpose

- "function" implies a definite end or purpose that the one in question serves or a particular kind of work it is intended to perform.

Activity natural or normal function: as

- a processus (as digestion) that an organism carries on or participates in by virtue of being alive
- a similar processus actually or potentially involving mental function; specifically : an educational procedure designed to stimulate learning by firsthand experience

Task

- a usually assigned piece of work often to be finished within a certain time.
- "task" implies work imposed by a person in authority or an employer or by circumstance.
- Synonym : duty, job.

Procedure

- a particular way of accomplishing something or of acting
- a series of steps followed in a regular definite order
- a traditional or established way of doing things

Steps

- a stage in a process
- an action, proceeding, or measure often occurring as one in a series

En conclusion, si l'on considère ces définitions, un processus est une suite d'activités et de fonctions, une fonction et un processus impliquent tous deux un résultat, une activité est un processus et implique l'utilisation de fonctions, une étape divise un processus en plusieurs morceaux, et pour compliquer le tout, un ensemble d'étapes forme une procédure.

Tout ce qu'on peut en déduire, c'est qu'il est question de travail divisé en étapes et assigné à un acteur (un employé). Les définitions de processus, activité, procédure et fonction se chevauchent les unes les autres.

2.3 Les définitions extraites du jargon informatique

Les définitions du dictionnaire se sont montrées insatisfaisantes, mais ces termes sont également utilisés dans le domaine informatique [1]. Voici quelques unes des définitions trouvées.

Process The sequence of states of an executing program. A processus consists of the program code (which may be shared with other processus which are executing the same program), private data,...

Function

- Computing usage derives from the mathematical term but is much less strict. In programming (except in functional programming), a function may return different values each time it is called with the same argument values and may have side effects.
- a computer subroutine; specifically: one that performs a calculation with variables provided by a program and supplies the program with a single result. (Merriam-Webster)

Procedure A procedure is a function which returns no value but has only side-effects.

En conclusion, les processus sont des programmes, qui peuvent incorporer des fonctions et des procédures, les fonctions sont elles mêmes des processus, ce qui dans le domaine des business processus est une aberration.

2.4 Les termes utilisés lors de la conception d'un processus

Les analystes et concepteurs de processus ont dû construire leurs propres termes liées à leur métier. Le maître mot pour les analystes et les (re-)concepteurs de processus est de ne pas confondre processus et fonction d'une organisation. Les définitions des dictionnaires ne satisfont pas cette exigence, ni les définitions liées au jargon informatique. Les définitions qu'ils ont mises au point ont été construites de manière à les aider dans leur travail, et sont centrées sur l'organisation.

2.4.1 Pourquoi ont-ils revus leurs termes?

Le premier problème qu'ont les concepteurs est tout d'abord de retrouver un processus dans environnement organisationnel fonctionnel pour l'analyser. Quand un projet de (re-) conception de processus pose problème, c'est en général parce que le processus visé ou sa portée n'ont pas été correctement identifiés au départ, ou qu'un sous-processus n'a pas été bien mis en contexte avec les autres. Parfois, les problèmes ressortent dès la planification de l'enchaînement des travaux, ils peuvent aussi apparaître seulement à l'implémentation quand les performances sont testées.

Voici 3 problèmes très généraux trouvés dans la conception d'un processus :

1. Les processus ne sont pas bien identifiés par les analystes. Les processus traversent les limites fonctionnelles d'une organisation, mais les analystes ont le modèle fonctionnel tellement ancré qu'ils retravaillent le processus de chaque département d'une organisation puis les enchaînent. Cela n'offre pas les mêmes gains, les mêmes avantages que la conception d'une processus global faisant intervenir tous les départements de l'organisation.
2. Les analystes n'arrivent pas à tracer un "chemin" d'un bout à l'autre du processus analysé sur un seul diagramme, car il est très largement étendu. Devant

l'étendue des tâches, une solution utilisée consiste à séparer des ensembles de tâches (formant des sous processus) en catégories, caractérisées par la fréquence d'utilisation de ces tâches (des transactions plusieurs fois par jour, une fois par semaine, ou par mois...).

3. Les processus sont améliorés par ceux qui les exécutent, ce qui entraîne une grande efficacité dans les limites d'un de ces processus, mais pas pour le client qui invoque ce processus, car les conditions à l'entrée de ce processus favorisent son exécution mais occasionnent des efforts et délais supplémentaires pour le client.

Pour que les projets de (re-) conception de processus puissent bien se dérouler et que ces problèmes soient évités, une bonne chose serait de bien définir le terme processus, ainsi que les tâches, fonctions, activités, procédures, étapes... tous ces termes ont des significations "courantes" qui peuvent se chevaucher. Elles doivent être considérées dans le contexte organisationnel et se distinguer.

2.4.2 La définition des (re-) concepteurs de business process

Un processus est une manière pour l'entreprise d'organiser le travail et les ressources (les employés, les informations) afin d'atteindre ses buts. Le travail et les ressources sont traditionnellement organisées en spécialités et en fonctions. Les organisations essaient maintenant d'organiser le travail et les ressources de manière à atteindre un résultat spécifique pour un client spécifique. Ils donnent à cette approche processus cette définition [11] : **"A business process is a collection of interrelated work tasks, initiated in a response to an event, that achieves a specific result for the customer of the process"**.

"En vue d'obtenir un résultat spécifique" : c'est la raison même de l'existence d'un processus. Les résultats peuvent être des biens, comme un produit correspondant à une commande, ou un service, comme une réponse à une requête. Le seul point important est que le résultat doit être clairement identifiable et dénombrable. Le processus "Développer un Nouveau Produit", "Remplir une Commande", "Résoudre un Problème Interne" répondent à ce point. On peut identifier chacun des produits développés et les dénombrer. On peut donc compter le nombre de fois que le processus "Développer un nouveau Produit", "Remplir une Commande" ou "Résoudre un Problème Interne" a été exécuté. Cependant, on ne peut compter le nombre de "Recherche et Développement", de "Help Desk", de "Marketing", de "Ressources Humaines" car ce sont des départements ou des fonctions, pas des processus. Un processus a un début, une fin et un résultat.

"Pour le client de ce processus" : le client peut être une personne, une organisation, par exemple un employé qui a appelé le processus "Résoudre un Problème Interne". Le point important est qu'il peut transmettre un jugement sur la qualité du processus et son résultat. Adopter la perspective du client aide à identifier et nommer ce processus. Par exemple, le service "état civil et population" délivre au client une carte d'identité. Un processus nommé "Traiter un formulaire de demande" ne

convient pas au point de vue du client, car le formulaire n'est pas ce en quoi le client est intéressé. Le fait que le formulaire soit rangé, analysé, ou envoyé à un autre service n'intéresse pas le client, il s'attend à recevoir une date à laquelle sa carte sera disponible pour venir la chercher.

"Déclenché en réponse à un événement spécifique" : on doit pouvoir suivre la trace d'un processus depuis son origine, c'est-à-dire l'événement qui l'a déclenché. Par exemple, l'événement déclencheur du processus "Développer un Nouveau Produit" est "une opportunité de marché est confirmée", qui peut être considéré comme une requête pour le développement d'un nouveau produit. Parfois, l'identification de l'événement déclencheur demande réflexion. Par exemple, le processus "Remplir la Commande d'un Client" est-il initié par la réception d'une commande client par l'organisation, ou par la demande du client proprement dite? Dans d'autres cas, plusieurs événements sont nécessaires pour déclencher un processus. Dans tous les cas, l'effort d'identification des événements en vaut la peine, il permet de mettre en relation les événements qui déclenchent un processus et le résultat de celui-ci.

"Des tâches" : un processus est un ensemble de tâches. Ici, dans cette définition, il n'y a aucune différence entre une tâche, une action, une activité, une étape, ou d'autres termes de ce genre, ils sont tous synonymes. Il s'agit de travail, effectué à un moment donné par un ou plusieurs acteurs. Un acteur peut être une personne jouant un rôle spécifique ou exerçant son activité professionnelle au sein de l'organisation, ou encore un département ou un système automatisé.

"Un ensemble de tâches en relations les unes avec les autres" : dans un processus, les tâches sont reliées selon un flux, la fin d'une tâche entraînant une autre tâche. Les départements et les autres acteurs effectuent des tâches l'une après l'autre, mais ces tâches n'ont en commun que celui qui les exécute, elles n'appartiennent pas au même processus, et ont été déclenchées par deux événements différents.

2.5 Les définitions de la WfMC

La WfMC (Workflow Management Coalition) est un organisme composé de fabricants de produits liés aux workflows : les Workflow Management Systems (WFMS). Ils travaillent ensemble de manière à émettre des standards et des modèles de référence pour leurs produits de manière à permettre l'interopérabilité de ceux-ci.

Les définitions émises par ce WfMC concernent les développeurs de WFMS et sont donc très orientées "technologies de l'information". Les concepteurs de processus qui ne s'intéressent qu'à l'organisation comprennent les termes comme business process, activité, transition, condition de transition, application (qu'il faut lancer pour accomplir une activité). Mais ils vont certainement ignorer la signification de certains termes qui sont strictement liés à l'organisation des données dans le WFMS : les différents types d'activités (un terme comme "activité de routage" est clair pour un développeur de WFMS mais pas pour un concepteur de processus : il ne mettra jamais une activité de routage dans son business process diagram !), des fonctions des différentes

applications utilisées, du traitement des différents types de données (les données de contrôle, les données du workflow, les données de applications) entre les activités.

Comme ce mémoire traite de la conception et l'implémentation d'un Workflow Management System, c'est bien sûr leurs termes et standards qui seront le plus pris en compte dans ce travail.

2.5.1 Que sont les Workflow Management Systems?

Avant l'arrivée du workflow management dans les organisations, un superviseur, responsable d'un (ou d'une partie d'un) processus, affectait les différentes activités de ce processus à des employés, sur base de leur formation, leurs connaissances, leur qualification ou encore leur expérience, et dressait des listes pour savoir où en était un travail bien précis en cours pour le faire avancer et mesurer les performances des employés effectuant ce travail.

Depuis, des outils ont été développés pour gérer les processus et leurs activités en remplaçant ces listes. Un Workflow Management System (WFMS) automatise, gère le processus en affectant les différentes tâches d'un travail aux employés, transmet les informations liées à ce travail d'un employé à un autre et surveille la progression du travail par rapport au processus général. Il enregistre une définition de la planification des traitements (réalisées par des acteurs appartenant aux organisations) et gère les informations qui y sont liées.

Les fabricants de logiciels proposent aujourd'hui une grande variété de produits sur le marché. Chacun de ces produits se focalise sur certaines fonctionnalités et les utilisateurs ont adopté l'un ou l'autre produit qui répondait le mieux à leurs besoins. Toutefois, des standards doivent être définis pour permettre à tous ces systèmes de workflow hétérogènes de travailler ensemble, pour ne pas se limiter à des systèmes d'automatisations de processus incompatibles entre eux.

Tous les produits de workflow management ont des caractéristiques communes, ce qui pourrait leur permettre d'atteindre un niveau d'interopérabilité s'ils utilisaient des standards communs pour certaines fonctions. Le WFMC, un groupe d'entreprises réunies pour créer ces standards, a identifié ces fonctions et a développé des spécifications pour le développement de produits liés aux workflows. Ces spécifications permettent de rendre différents systèmes de management de workflow inter-opérables et permettent une meilleure intégration des applications existantes dans l'organisation avec le système de management de workflow.

2.5.2 La terminologie du WfMC

Le WfMC a publié beaucoup de documents sur les workflow et a construit une terminologie [15] utilisée dans tous ses documents et ses produits, pour permettre une cohérence entre ceux-ci. Malheureusement, la définition de ces termes, les documents publiés, et les produits du WfMC ne sont pas encore partagés par tous les fabricants de produits de workflow. Voici la définition que donne le WfMC du business process ou processus d'une organisation.

Définition - processus d'une organisation Un processus d'une organisation (business process) est un ensemble d'une ou plusieurs activités reliées l'une à l'autre pour réaliser ensemble un objectif d'une organisation définissant des rôles, des fonctions et une relation avec un client déclenchant ce processus. Cette définition appelle quelques remarques :

- Un processus organisationnel est associé à un objectif organisationnel et un client. Il peut être contenu dans une fonction de l'organisation ou s'étendre à travers plusieurs organisations, reliées entre elles par une relation client - fournisseur. Un processus organisationnel a des conditions qui déclenchent le démarrage de chacune de ses instances et s'achève en produisant un output bien défini.
- Un processus organisationnel comprend des interactions formelles et informelles entre les participants, la durée de ces interactions peut varier grandement selon les cas.
- Un processus organisationnel consiste en des activités automatisées (qui peuvent être supportées par le WFMS) et des activités manuelles (qui se trouve en dehors des limites du WFMS).

La définition est équivalente mais certains termes sont différents, par exemple "activité" a remplacé "tâche". C'est souvent le cas, différents produits de workflow adoptent leurs propres termes et leurs propres définitions. Des termes comme "activité", trop souvent utilisés, finissent par avoir plusieurs sens selon le produit ou le milieu considéré (donc plus aucun sens concret et utilisable), à un tel point que certain parlent de bannir certains terme [6]: "les définitions de processus sont en général exprimées en terme d'activités ... le terme activité fausse la distinction entre un état et une action. Une état (d'attente) dans un processus souligne une dépendance à un acteur extérieur. En cours d'exécution, cela signifie que le moteur de workflow doit attendre que l'acteur externe dise au système que l'état est fini. Une action est un programme que le système exécute quand un événement spécifique apparaît en cours d'exécution ... éviter le terme activité et le remplacer soit par état, soit par action." Ici, pour certaines raisons, on suggère que le terme activité soit remplacé non pas par tâche, mais par état ou action.

Pour éviter le problème qu'apportent les définitions des termes liées au workflow à travers la communauté du workflow, les termes utilisés dans ce mémoire seront ceux utilisés par le WfMC.

2.5.3 Qu'est ce qu'un workflow?

Définition - Workflow Un workflow (appelé parfois aussi workflow management, workflow computing ou encore case management) est l'automatisation (partielle ou entière) informatisée d'un processus d'une organisation (business process) au cours duquel des informations sont passées d'un participant à un autre, selon un ensemble de règles procédurales. Il faut noter que :

- l'automatisation d'un processus organisationnel est définie dans le cadre d'une définition de processus qui identifie les différentes activités, les règles procédurales et les données de contrôle servant à caractériser l'état de l'instance d'un processus pendant son exécution (arrêté, en cours d'exécution, fini,...).

- plusieurs instances de processus peuvent exister pendant l'exécution d'un processus. Un ensemble spécifique de données est lié à chacune de ces instances.
- une distinction est parfois faite entre les workflows de production et les workflows ad-hoc. Dans un workflow de production, la plupart des règles procédurales sont faites à l'avance, alors que dans un workflow ad-hoc elles peuvent être créées ou modifiées pendant que le processus est en train de s'exécuter.

On parle de workflow dans le contexte de systèmes fournissant un support pour l'automatisation des business processes. C'est dans ce domaine que la WfMC travaille. Le Business Process Reengineering débouche sur une implémentation du business process, mais la définition des processus est séparée de la conception et l'implémentation du système de gestion de workflow. L'implémentation du système comprend des technologies non abordables par les concepteurs de business process, mais la séparation du système et de la définition des processus leur permet justement des changements à leur processus (amélioration continue, Kaizen) sans toucher au système de gestion de workflow. La définition du processus devient un programme exécuté par un Workflow Management System, c'est-à-dire un workflow.

Un Workflow Management System est un système fournissant l'automatisation du côté procédural des processus d'une organisation, par la gestion de sa séquence d'activité et la gestion de l'implication des ressources humaines ou informatiques appropriées à chacune de ces activités.

Définition - Workflow Management System (WFMS) Un WFMS (on parle aussi de automatisation de workflow, gestionnaire de workflow ou système informatique de workflow) est un système qui définit, crée et gère l'exécution de workflows en utilisant des logiciels, en exécutant un ou plusieurs moteurs de workflow, en étant capable d'interpréter une définition de processus (une description formelle d'un business process), et étant capable d'invoquer des applications logicielles et des d'autres outils si nécessaire. On peut remarquer que :

- un workflow management system consiste en un ensemble de composants permettant d'interpréter des définitions de processus, créer et gérer des instances de processus en cours d'exécution, contrôler les interactions avec les participants et les applications. De tels systèmes proposent également des fonctions d'audit pour chaque instance de processus et d'informations administratives générales sur le système entier.
- la WfMC a publié un modèle de référence (le workflow reference model) définissant l'architecture, la structure et les interfaces d'un WFMS.

De tels systèmes peuvent être implémentés de différentes manières, selon les technologies choisies et l'environnement (dans une entreprise ou entre plusieurs entreprises). Le workflow reference model prend donc une vision très large de la gestion des workflows, pour pouvoir couvrir le plus grand nombre possible de choix que pourrait faire une entreprise pour l'implémentation d'un tel système.

2.5.4 Pourquoi utiliser des Workflow Management Systems?

Quels sont les cas où l'utilisation d'un WFMS pourrait être bénéfique? Voici trois cas où le WFMS peut procurer des avantages par son utilisation [6].

L'Enterprise Application Development Les WFMS sont utilisés pour l'Enterprise Application Development (EAI), qui est l'implémentation de logiciels constitués de plusieurs applications dédiées. Un WFMS peut aider cette discipline.

Prenons l'exemple d'une organisation qui dépend d'applications hétérogènes, ayant leurs propres bases de données. Chacune de ces applications ont leur propre fonction (la gestion des clients, la gestion des documents, la réception des commandes, la facturation). Appelons-les "applications dédiées". Elles contiennent toutes des traitements métiers, donc une partie du business process dans leur code, qui ne peuvent être modifiés. L'organisation arrange ces parties éclatées, ces mini-process, entre eux en un grand business process non automatisé.

Les applications dédiées contiennent donc dans leur code des traitements métiers qui forment une partie immuable du processus principal et une connaissance du domaine. Au contraire, un Workflow Management System n'a aucune connaissance du domaine, mais est plus flexible, car la définition du processus peut-être modifiée.

Si on utilise uniquement un WFMS, les traitements métiers contenus dans les applications dédiées doivent être réécrits dans une description de processus pour que le WFMS "comprenne" le domaine. C'est pourquoi les applications dédiées et les WFMS sont complémentaires. Si une organisation achète une application dédiée, elle achète en même temps des business processes fixes codés dans cette application. On utilise un WFMS pour automatiser le processus entier, les sous-processus sont implémentés par les applications dédiées, si ceux-ci conviennent.

Un moteur de workflow embarqué dans une application En partant à nouveau du fait que les applications dédiées comprennent dans leur code une partie d'un processus de l'organisation, on peut imaginer que ces applications comprennent non pas du code immuable mais un système d'exécution de workflow embarqué. Cela permettrait une facilité de réutilisation et de maintenance pour l'application.

Le développement itératif des processus Les organisations passent beaucoup de temps dans l'analyse de leur business process dans l'espoir d'obtenir un résultat stable et durable. Malheureusement, les processus nécessitent souvent des améliorations après leur mise en place. Des démarches d'amélioration de processus sont un comportement normal pour toute organisation.

Plus la logique de leur processus est étalée dans des applications dédiées et immuables, plus changer de processus sera difficile et cher. Les organisations seront tentées de modifier la manière de se servir des applications plutôt que de modifier le processus contenu dans l'application. Un WFMS permet de changer plus facilement de processus sans entrer dans le code d'une application dédiée.

2.5.5 Les différentes catégories de fonctions d'un WFMS

A un haut niveau, tous les WFMS reprennent trois grandes catégories de fonction [7] (figure 2.1):

1. les build-time functions, qui comprennent la définition, parfois la modélisation, du workflow du processus et des activités qui le constituent,
2. les run-time process control functions, qui concernent la gestion des workflows lors de leur exécution et la gestion de la succession des différentes activités constituant chaque instance de processus,
3. les run-time interactions, les fonctions d'interaction avec les participant (les acteurs, utilisateurs) et les outils (applications) pour effectuer les travaux liés aux activités.

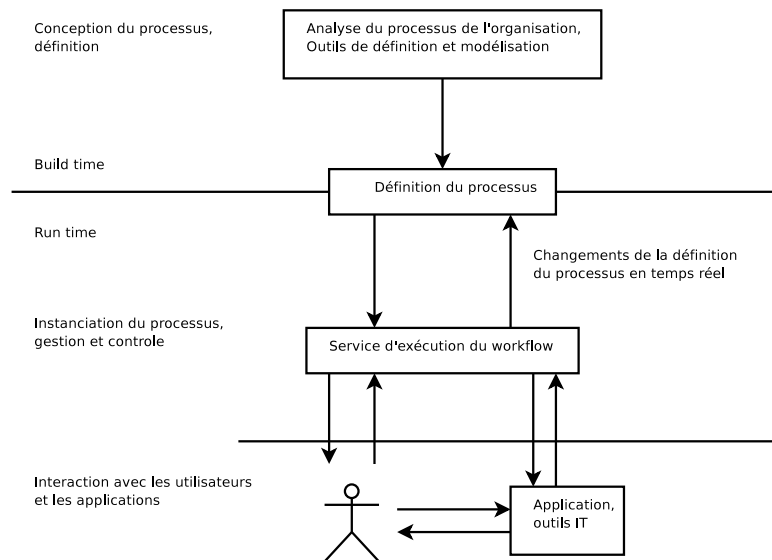


FIG. 2.1 – Représentation schématique des fonctions d'un WFMS.

2.5.5.1 Les build-time functions - les fonctions de définition de processus

Les build-time functions forment la première catégorie de fonction d'un WFMS. On définira, outre cette catégorie de fonctions, différents concepts qui leur sont liés : le processus (tout court), la modélisation de processus, la définition de processus, l'activité (automatisée ou manuelle).

Les build-time functions sont celles qui donnent comme résultat une définition (exprimée dans un langage informatique, par exemple, comme c'est le cas dans ce mémoire, un langage dérivé de XML) du processus d'une organisation. C'est pendant cette phase que le processus organisationnel du monde réel est analysé, modélisé et exprimé dans le langage de définition de processus utilisé par le WFMS. Suite à la

définition des build-time functions, il y a donc trois premiers concepts à définir : le processus, la modélisation de processus et la définition de processus.

Définition - Processus Un processus (ou encore réseau d'activités, graphe, réseau de pétri, modèle, feuille d'instructions) dénote la vision formalisée d'un processus organisationnel (business process), représenté sous forme d'activités coordonnées en parallèle ou en série, connectées entre elles pour atteindre un but commun.

Définition - Modélisation du processus La modélisation du processus (on parle aussi de mode de définition de processus, de modélisation de processus organisationnel ou de période de construction) est l'activité pendant laquelle la description des activités manuelles ou automatisées d'un workflow sont définies ou modifiées. On notera que les définitions de processus sont initialement définies avant l'exécution du workflow, elles peuvent être modifiées plus tard ou pendant l'exécution du processus, sous des conditions ou privilèges spéciaux, selon un rôle spécifique accordé à un participant.

Définition - Définition de processus Une définition de processus (appelé aussi diagramme de flux, définition du modèle, définition du routage, digramme état transition, ...) est la représentation d'un processus organisationnel (business process) sous une forme qui permet des manipulations automatisées, comme la modélisation (grâce à un outil logiciel) ou l'exécution par un WFMS. La définition d'un processus consiste en un réseau d'activités et leurs relations, des critères pour indiquer le début et la fin du processus, et des informations sur les différentes activités, comme les participants, les applications associées aux activités, les données, etc. On peut remarquer aussi que :

- La définition du processus résulte de l'activité de modélisation de processus, celui-ci comprend des activités manuelles et automatisées.
- La définition de processus peut contenir des références à des sous-processus, définis séparément, qui représentent une partie de la définition complète.
- La définition de processus peut faire référence à des organisations et des ressources externes, des participants peuvent être indirectement définis.
- Le WfMC reference model comprend une interface pour l'importation et l'exportation de définitions de processus. Il incorpore le méta-modèle de définition de processus, qui identifie les entités de haut niveau de la définition du processus.

Le processus doit donc être défini de manière textuelle dans un langage formel, mais peut être aussi exprimé sous une notation graphique. La WfMC ne considère pas que la création de la définition du processus en utilisant des outils graphiques soit sujet à standardisation, et trouve donc normal que différentes notations existent sur le marché. Aucune notation graphique des processus n'est donc standardisée par la WfMC. Par contre, le résultat des build-time functions, la définition du processus dans le langage utilisé par le système, est identifié comme devant être standardisé pour permettre l'échange des données liées aux définitions de processus entre différents outils build-time (outils de conception d'un processus) et des outils run-time (outils qui fournissent un environnement d'exécution à ce processus).

La définition de processus consiste en un réseau d'activités. Les activités peuvent être soit manuelles, soit automatisées. Certaines activités peuvent être aussi de "fausses

activités", uniquement présentes dans le but de structurer la définition de processus et le rendre plus lisible.

Définition - Activité Une activité (appelée aussi étape, noeud, tâche, opération, instruction) est la description d'un travail formant une étape logique d'un processus. Une activité peut être manuelle (ne supportant pas une automatisation) ou automatisée. Elle nécessite des ressources (humaines ou informatiques) pour supporter l'exécution du processus, dans lequel elle est assignée à un participant. Par ailleurs, on peut noter que :

- la définition d'un processus consiste en un ensemble d'activités qui sont reliés logiquement pour contribuer à la réalisation du processus complet.
- une activité est typiquement la plus petite unité de travail programmée dans un moteur de workflow pendant l'exécution du processus (caractérisé par une transition et des conditions pré et post). Toutefois, une activité peut comprendre plusieurs éléments de travail (anglais : work items) pouvant être affectés à un participant.

Définition - Activité automatisée Une activité automatisée est une activité entièrement gérée par le WFMS pendant l'exécution du processus organisationnel dans lequel elle prend part. Pendant l'exécution du processus, une activité est gérée par le WFMS, cela peut donner en pratique :

- une application qui est directement invoquée par le WFMS et qui se déroule sans l'intervention d'un participant humain.
- un ou plusieurs éléments de travail (work items) affectés à un participant, ces éléments impliquent l'utilisation d'applications et d'outils invoqués et gérés par le WFMS.
- un ou plusieurs éléments de travail affectés à un participant qui exécute ces éléments de façon indépendante du WFMS. La terminaison de ces éléments de travail est notifiée au WFMS par le participant. Certains WFMS appellent cela une activité manuelle.

Définition - Activité manuelle Une activité manuelle (que l'on appelle aussi activité non-automatisable, étape manuelle, tâche humaine, travail manuel) est une activité d'un processus organisationnel qui n'est pas automatisable et qui se situe dès lors hors de la portée du WFMS. De telles activités peuvent être mentionnées dans la définition d'un processus (lors de la modélisation), mais ne font pas partie du workflow (c'est-à-dire la partie automatisée du processus organisationnel) résultant.

2.5.5.2 Les run-time process control functions - les fonctions de contrôle de l'exécution des processus

Cette section décrit les run-time process control functions forment la seconde catégorie de fonctions d'un WFMS. En plus de définir ce que sont ces fonctions, de nouvelles définitions, liées à l'exécution d'un processus, feront leur apparition. Ces

fonctions concernent les processus en cours d'exécution. On parlera donc ici d'exécution de processus, d'instance de processus et d'instance d'activité.

Définition - Exécution du processus L'exécution du processus est la période pendant laquelle le processus est opérationnel, autrement dit quand des instances de ce processus sont créées et gérées.

On parle d'exécution de processus pour bien différencier cette phase de la phase de modélisation (ou de définition) du processus, pendant laquelle la structure du processus est définie. Toutefois, dans le cas où la définition du processus est modifiée en cours d'exécution, cette appellation est sans objet.

A l'exécution, les définitions de processus sont interprétées par un système de contrôle (appelé workflow enactment service ou service d'exécution de workflow) responsable de la création et du contrôle des instances de processus, planifiant les différentes instances d'activités au sein de chaque instance de processus et invoquant les ressources humaines et applications nécessaires. Ces fonctions de contrôle du processus en cours d'exécution forment le lien entre le processus tel qu'il est modélisé dans sa définition et le processus en exécution, reflété par les run-time activity interactions (voir plus loin) avec les participants et les applications. L'élément central d'un WFMS est le "moteur de workflow", qui est l'outil permettant la création des instances de processus et leur destruction, la succession des activités et les interactions avec les participants et les applications. Les termes "service d'exécution de workflow" et "moteurs d'exécution de workflows" seront détaillés et définis ultérieurement.

Définition - Instance (d'un processus ou d'une activité) Une instance est la représentation d'une exécution (particulière) d'un processus ou activité d'un processus et de ses données associées. Chaque instance représente un thread (différencié, unique) d'exécution du processus ou de l'activité, qui peut être contrôlé de façon indépendante et aura son propre état interne et sa propre identité, et peut être utilisé comme un outil pour insérer ou récupérer des informations d'audit liées à cet exécution. De plus :

- un thread différencié d'exécution du processus ou d'activité : quand un processus implique différentes activités exécutées en parallèle, une instance de processus peut inclure des threads d'exécution multiples et concurrents (routage parallèle, and-split, and-join)
- une instance d'un processus ou d'une activité est créée et gérée par le WFMS pour chaque invocation de ce processus ou de cette activité.

2.5.5.3 Les run-time activity interactions

Les différentes activités d'un processus sont liées à des participants et réalisées avec l'aide d'outils particuliers, ou sont liées à des applications manipulant certaines informations (par exemple, mettre à jour une base de données). Des interactions entre ces participants, ces applications et le système de contrôle sont nécessaires pour que se succèdent les activités, pour valider l'état d'un processus, pour invoquer les applications et leur faire passer certaines données, etc.

Définition - Participant Un participant (ou encore acteur, agent, utilisateur, joueur de rôle, travailleur) est une ressource qui effectue le travail représenté par une instance d'activité. Ce travail se divise normalement en plusieurs éléments de travail attribués au participant via la liste de travaux. Notons aussi que :

- le terme "participant" s'applique normalement à une ressource humaine mais pourrait conceptuellement s'appliquer à une machine (agent automatisé ou agent intelligent).
- un participant peut être identifié directement depuis la définition du processus organisationnel, ou (plus probablement) identifié par référence, depuis la définition du processus, à un rôle ou une unité organisationnelle. Ce rôle peut être rempli par une ou plusieurs ressources disponibles, sélectionnées par le WFMS lors de l'exécution du processus.
- dans le méta-modèle de la définition de processus, la déclaration du participant peut identifier celui-ci comme étant un humain, une machine, un rôle ou une unité organisationnelle. Elle peut également faire référence à un modèle d'organisation externe (le modèle d'une organisation).

On peut tirer beaucoup de bénéfices à standardiser les interfaces de telles interactions. Une interface commune à plusieurs WFMS permettrait d'utiliser les mêmes applications avec ces différents produits.

2.5.6 Les interfaces d'un WFMS

Une des plus grandes fonctionnalités d'un WFMS est de distribuer le travail à des participants pendant l'exécution. Le service d'exécution de workflow fournit des interfaces pour récupérer les travaux des participants.

Le flux de travail peut impliquer le transfert des différents travaux vers d'autres systèmes d'exécution de workflow, permettant aux activités d'un même processus de s'exécuter sur différents services d'exécution de workflow. Des standards permettant de contrôler ces transferts de workflows permettent aux applications d'utiliser plusieurs produits de workflows comme si il s'agissait d'une seule entité logique.

Les interfaces standard que la WfMC ont publié sont donc :

1. la spécification des données liées aux définitions de processus, l'importation et l'exportation de ces définitions
2. des interfaces permettant l'interopérabilité entre les différents WFMS
3. des interfaces pour permettre les interaction avec les applications (outils IT)
4. des interfaces pour les fonctions utilisateur
5. des interfaces pour l'obtention de rapports de monitoring, métriques.

2.6 Conclusion

Nous avons vu au chapitre 1 que les workflows ne sont pas encore entrés dans leur phase de maturité. En conséquence, les concepts et les définitions liées à ce domaine n'ont pas encore été universellement reconnues et acceptées par l'ensemble des acteurs impliqués par les business processes et les workflows. Pour y remédier, ces acteurs

se sont groupés en organisations de standardisation (par exemple la WfMC) pour permettre au domaine de disposer d'informations de référence. Celles-ci concernent non seulement les concepts et leur définition, mais également les fonctionnalités et la structure générale d'un produit de gestion et d'exécution de workflow.

Ce chapitre a montré le lien entre un business process et un workflow. Il a aussi présenté les premiers termes utilisés dans le monde des business processes et des workflows à travers la présentation des fonctions de base d'un workflow management system. Sur base de ces premiers concepts définis, nous pouvons maintenant, dans les chapitres suivants, explorer plus en détail la structure d'un workflow management system et continuer l'exploration du domaine vers concepts plus spécifiques.

Chapitre 3

Le Modèle de référence des Systèmes de Gestion de Workflow

Le chapitre 1 proposait une introduction aux business process et workflows, il présentait les origines du BPM (Business Process Management) et des workflows. Le chapitre 2 établissait le lien entre business process et workflow et avait défini les termes de base utilisés dans le monde des workflows.

Le chapitre 3 propose la présentation du modèle de référence pour les systèmes de gestion de workflow, établi par la WfMC (Workflow Management Coalition). Ce modèle de référence et les concepts qu'il introduit et définit seront utiles dans l'analyse du domaine lors du dernier chapitre de ce mémoire, la conception d'un WFMS.

Ce chapitre sur le modèle de référence des systèmes de workflow étend le précédent chapitre, car le modèle est basé sur la structure générique d'un WFMS (Workflow Management System) présentée à la fin du chapitre 2 et la liste des définitions des termes de base commencée au chapitre 2 se poursuit ici avec des concepts plus spécifiques aux WFMS.

Ce chapitre commence par présenter l'organisme qui a émis ce modèle de référence : la WfMC. Une brève introduction rappelle ses missions, son travail et ses résultats.

La première section présente la base du modèle de référence des systèmes de gestion de workflow : la structure générique d'un WFMS.

La deuxième section présente le Wf Reference Model, le modèle de référence pour les systèmes de workflows émis par la WfMC, ainsi qu'une définition de tous les concepts qui y sont attachés, de manière à obtenir une analyse du domaine la plus complète possible.

La troisième section présente un autre standard émis par la WfMC et lié au modèle de référence : les cinq interfaces qui relient les différents composants d'un système de gestion de workflow (une application de modélisation de processus, un gestionnaire de liste de travaux, les outils d'administration, ou un autre service d'exécution de workflow) avec le composant principal du WFMS : le système d'exécution de workflow.

3.1 Introduction

La Workflow Management Coalition a été fondée en 1993. Comme nous l'avons dit précédemment, c'est une organisation internationale sans but lucratifs composée de groupes de fabricants de produits liés aux workflows, d'utilisateurs de ces produits, d'analystes et de chercheurs. Les membres qui la composent sont plus de 300 : la WfMC est le plus gros émetteur de standards (mais toutefois pas le seul) dans le domaine des workflows.

Sa mission est la suivante :

- promouvoir et développer l'utilisation des produits de workflows
- diminuer les risques liés à l'utilisation des produits de workflows
- faire connaître les technologies de workflows, étendre le marché

La WfMC a établi un certain nombre de groupes de travail, chacun travaillant sur un domaine particulier de spécifications. Les groupes de travail sont structurés de la même manière que le workflow reference model, c'est-à-dire selon cinq interfaces. A chaque interface correspond un groupe de travail. Les groupes se réunissent individuellement mais aussi ensemble en un comité technique, responsable de la direction des travaux et de la coordination.

Le groupe de travail 1 s'occupe de l'interface 1 qui concerne l'échange (importation et exportation) de définitions de processus. Il fournit un méta-modèle pour décrire une définition de processus et un schéma XML décrivant le langage XPDL (XML Process Definition Language), un langage de description de processus de référence.

Le groupe de travail 2 fournit les spécifications pour les applications clientes utilisées pour communiquer avec les systèmes de workflows.

Le groupe de travail 3 concerne les applications invoquées par les systèmes de workflows, lors de l'exécution des activités. Ce groupe a été fusionné avec le groupe 2.

Le groupe de travail 4 est chargé de l'interface 4, l'interopérabilité entre produits de workflows. Les standards émis sont ASAP (Asynchronous Web Service Protocol) et Wf-XML.

Le groupe 5 reprend les données d'audit de l'interface 5.

La WfMC a d'abord publié le workflow reference model [7], définissant une architecture générale et une ensemble de termes et de définitions communs [15] pour l'industrie du workflow. Une étape importante a été la publication de la spécification de la Workflow API [14], couvrant les interfaces avec les différents composants, l'interopérabilité entre produits. Les informations d'audit et les spécifications d'importation et l'exportation de définitions de processus sont apparues un peu plus tard, ainsi que les spécifications pour l'invocation d'applications. La WfMC a également publié un langage de description de processus, XPDL (XML Process Description Language) [16], un langage de référence pour les descriptions de processus.

La WfMC a mis en pratique l'utilisation de ses spécifications par des démonstrations de prototypes, montrant des scénarios d'interopérabilité entre différents produits de workflow, selon les spécifications publiées.

3.2 Structure générique d'un WFMS

Connaissant les différentes implémentations des WFMS sur le marché, La WfMC a construit un modèle général d'implémentation qui peut décrire la plupart des produits et donc fournir une base commune pour les scénarios d'interopérabilité.

Le modèle identifie les composants fonctionnels principaux d'un WFMS et les relations entre ceux-ci. En pratique, ce seront des variantes concrètes de ce modèle d'implémentation qui seront sur le marché. Les fabricants de produits pourront éventuellement publier les interfaces de ces différents composants si celles-ci suivent des spécifications standard, leur donnant un degré de conformité plus ou moins grand, permettant à d'autres produits d'identifier les fonctions internes de leur système pour pouvoir les intégrer avec d'autres produits sur le marché.

Ce modèle générique a trois types de composants :

1. des composants remplissant des fonctions interne au système (outil de définition de processus, moteur de workflow, gestionnaire de liste de travaux, interface utilisateur),
2. des types de données liées aux définitions de processus et liée au workflow c'est-à-dire les données de contrôle du workflow, données relatives au workflow, liste de travaux, modèle organisationnel (rôle), définition de processus,
3. les application et les bases de données de ces applications, qui sont en dehors des limites du WFMS, mais qui peuvent être invoquées lors de l'exécution d'un processus par le WFMS.

Voici la structure générique d'un WFMS (figure 3.1). Les couleurs reflètent les composants du modèle : en rouge le outils, en bleu les données accessible par le moteur d'exécution de workflow, en jaune les applications (servant de support à une activité précise) et leurs données.

3.3 Le Workflow Reference Model

Le workflow reference model [7] a été développé au départ de la structure générique d'un WFMS au chapitre précédent. Le modèle identifie les interfaces au sein de cette structure en vue de permettre l'interopérabilité avec d'autres produits, et entre les composants génériques de cette structure. Un ensemble standardisé d'interfaces et de formats d'échange de données est donc nécessaire.

La figure 3.2 illustre les principaux composants et interfaces d'une architecture de WFMS.

Le modèle met en évidence les interfaces et les composants. Les interfaces seront considérées une par une dans les sections suivantes.

L'interface du service d'exécution de workflow est appelée WAPI (Workflow Application Programming Interface). Elle peut être considérée comme un ensemble de services par lesquels le système de workflow est accessible et qui définit les interactions entre le logiciel de contrôle du système et les composants externes.

Dans les fonctions définies par les cinq interfaces, il y en a qui appartiennent à plusieurs interfaces en même temps, donc il est mieux de considérer la WAPI comme

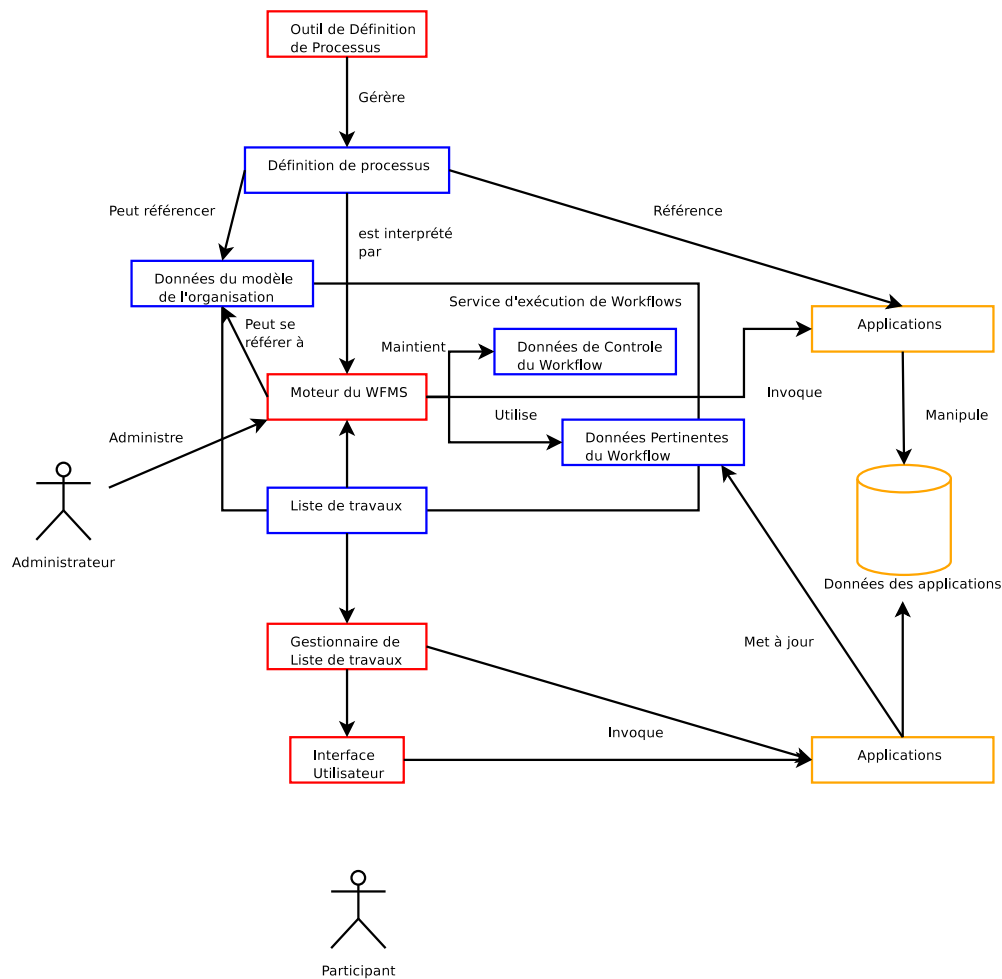


FIG. 3.1 – Structure générique d'un WFMS.

un service d'ensemble fournissant un service de management de workflow à travers cinq domaines fonctionnels plutôt que cinq domaines indépendants.

3.3.1 Qu'est ce qu'un service d'exécution de workflow ?

Le service d'exécution de workflow fournit un environnement d'exécution pour les instances de processus en utilisant un ou plusieurs moteurs de workflow, responsable de l'interprétation des définitions de processus et de l'exécution des instances de processus et d'activités. La raison d'être du service d'exécution de workflow est de permettre l'interopérabilité entre plusieurs moteurs. Si le système ne dispose que d'un seul moteur, alors le système d'exécution de workflow peut être assimilé au moteur de workflow.

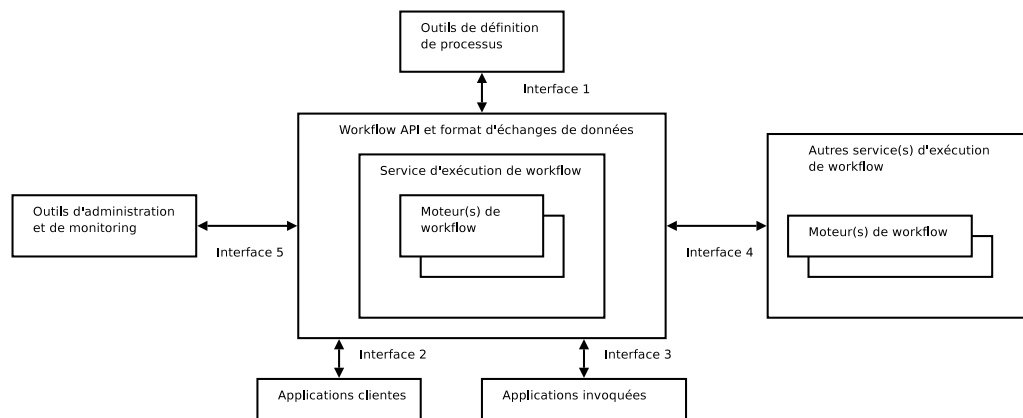


FIG. 3.2 – Les composants du modèle de référence des WFMS.

Définition - Service d'exécution de workflow Le service d'exécution de workflow est un service comprenant un ou plusieurs moteurs de workflow, et permet de créer, gérer et exécuter des instances de processus particulières. Les applications interagissent avec ce service via une partie de la WAPI. Un service d'exécution de workflow peut opérer dans un seul domaine de workflow (des moteurs de workflow homogènes) ou plusieurs domaines différents (moteurs de workflow hétérogènes).

Le modèle fait une séparation entre

1. le contrôle des activités et des processus (fourni par le service d'exécution de workflow),
2. les applications et les tâches des participants qui constituent le traitement associé à chaque activité.

Cette séparation fournit l'opportunité d'adoption de standards et d'intégration d'applications clientes pour un WFMS particulier :

1. L'interface des applications clientes (interface 2), par laquelle un moteur de workflow interagit avec un gestionnaire de liste de travaux, responsable de l'organisation du travail d'un participant particulier. Le gestionnaire de liste de travaux sélectionne des éléments de travail dans une liste de travail. L'activation d'outils (sous forme d'applications) peut être gérée par le gestionnaire de liste de travaux (la plupart du temps) ou par le participant lui-même.
2. L'interface des applications invoquées (interface 3), qui permet au moteur de workflow d'activer directement une application (typiquement sans interface utilisateur) pour traiter le travail correspondant à une activité particulière.

Le service d'exécution de workflow est considéré comme une seule entité logique mais peut comprendre des composants distribués, dans lequel plusieurs moteurs de workflow hétérogènes interagissent chacun avec un sous-ensemble des participants et des activités d'une même instance de processus. Pour que les définitions de processus et les noms de participants et d'applications doivent être gérées sur une même base, les

systèmes doivent avoir les mêmes conventions de nom pour ces éléments. L'interface 4 est responsable de l'échange d'information concernant les activités et les processus, leurs instances et leur données lors de l'exécution entre les différents moteurs de workflow.

3.3.2 Qu'est ce qu'un moteur de workflow?

La précédente section a introduit le système d'exécution de workflow, composé de moteurs de workflow. Cette section définit les responsabilités de ce moteur.

Définition - Moteur de workflow Un moteur de workflow (appelé aussi workflow management engine) est un logiciel qui fournit un environnement d'exécution pour une instance de processus. De plus :

1. Le moteur de workflow fournit des fonctions pour supporter l'exécution des (instances de) processus des organisations, basés sur les définitions de processus. Ces fonctions comprennent :
 - l'interprétation des définitions de processus,
 - la création des instances de processus et d'activité ainsi que la gestion de leur exécution, incluant des fonctions comme start / stop / suspend / resume / etc... : le moteur gère les transitions entre les états internes des instances de processus et d'activité,
 - les transitions entre les instances d'activités (routage séquentiel ou parallèle, gestion des différentes données associées au workflow) et la création d'éléments de travail appropriés pour leur traitement,
 - des interfaces vers les applications invoquées (les applications utilisées lors du travail défini par une activité) et les données relatives au workflow qui leur sont nécessaires,
 - des fonctions de supervision et gestion.
2. La gestion d'information sur la connexion et déconnexion de participants au système,
3. Le moteur de workflow exclut généralement les fonctions de gestion de la liste de travail, qui sont centrées utilisateur, quoique ils pourraient être intégrés au moteur,
4. Un ou plusieurs moteurs de workflow forment ensemble un domaine de workflow, qui fournit un environnement homogène d'exécution de processus. Un service d'exécution de workflow fournit le moyen d'exécuter des workflows spécifiques sur un ou plusieurs moteurs de workflows, qui peuvent faire partie de un ou plusieurs domaines de workflow,
5. Plusieurs moteurs de workflow peuvent coopérer pour se partager l'exécution de workflows (interopérabilité entre workflows).

3.3.3 Les transitions d'état des instances de processus et d'activités

La section précédente parlait des moteurs de workflow dont certaines des responsabilités étaient la création des instances de processus et d'activité, ainsi que les

transitions entre les différents états internes de ces instances. Cette section examine ce sujet plus en détail.

Définition - Transition d'état (interne) Une transition d'état est un mouvement d'un état interne (d'une instance d'activité ou de processus) vers un autre, reflétant un changement dans le statut du workflow. Par exemple, la création d'une nouvelle activité spécifique donne un état à cette activité et son commencement change encore son état, ainsi que l'état de l'instance du processus. Une transition d'état peut avoir lieu en réponse à un événement extérieur, par exemple un appel d'un participant via une API, une décision de routage prise par le moteur de workflow, etc ...

Notons que, pendant l'exécution du workflow, une série de transitions d'état a lieu et peut être présentée sous forme d'information d'audit.

Le service d'exécution de workflow est aussi une machine de transition d'état où une instance de processus et d'activité change l'état du système en réponse à des événements externes (par exemple la fin d'une activité) ou à une décision prise par le moteur de workflow (par exemple la transition vers un autre état d'activité au sein d'un processus). En ce qui concerne les instances de processus et les états internes de ces instances, voici ce que le modèle propose.

Définition - Instance d'un processus Une instance d'un processus (appelée aussi instance d'une définition de processus, instance d'une définition de workflow, case) est la représentation d'une exécution particulière d'un processus. On peut ajouter à cette définition les remarques suivantes :

- Une instance de processus est créée, gérée et finalement "terminée" par un WFMS, conformément à la définition de ce processus.
- Chaque instance de processus représente une exécution individuelle du processus, l'instance a ses propres données, peut être contrôlée de façon indépendante, et peut donner des informations d'audit au fur et à mesure qu'elle progresse vers son état final et sa terminaison. Elle représente le travail d'un processus organisationnel analysé et normalisé par un WFMS (par exemple, une réclamation de dommages à une assurance, l'invention d'un nouveau produit).
- Chaque instance de processus peut exposer son état interne, représentant sa progression vers sa terminaison, et son statut (voir état d'un processus) en concordance avec les activités qui constituent le processus.
- Certains processus ne se terminent jamais mais restent dans un état "dormant", archivé, de manière à respecter certaines exigences légales.

Définition - Etat d'un processus L'état d'un processus (ou encore état du workflow ou état du modèle) est la représentation de (conditions internes définissant) l'état d'une instance de processus à un instant bien déterminé. Les WFMS considèrent ces informations comme faisant partie des données de contrôle du workflow. Notons aussi que :

- L'état de chaque instance de processus en exécution est maintenu par le WFMS. Différents WFMS peuvent avoir différentes façons de représenter l'état d'un pro-

cessus et peuvent avoir leur propre ensemble d'état de processus et de définitions de ces états.

- Pendant l'exécution d'une instance de processus, cette instance peut prendre différents états séparés par des transitions (d'état). La définition du processus reprend l'ensemble des différents états que l'instance du processus peut prendre.
- L'interface WAPI définit des appels pour manipuler les informations sur l'état d'un processus, par exemple interroger le WFMS sur l'état d'un processus ou forcer la transition vers un nouvel état.

Voici les états qu'une instance de processus peut prendre, ainsi que les transitions entre ces états (figure 3.3). Cette liste est exhaustive.

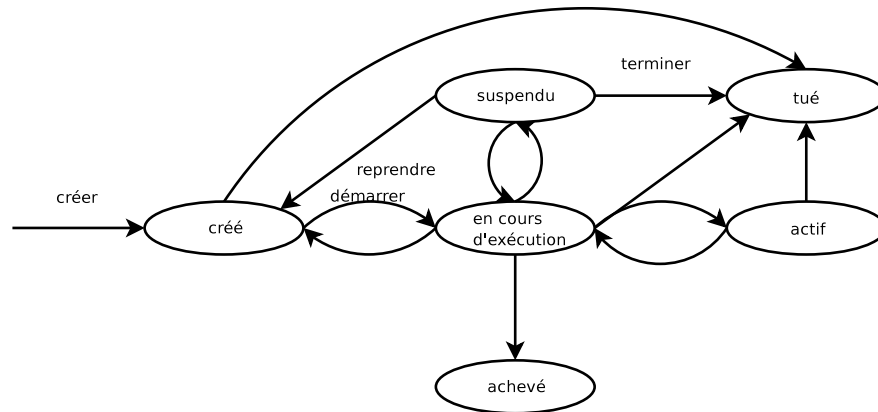


FIG. 3.3 – Les différents états qu'une instance de processus peut prendre.

- créé : l'instance du processus a été créée, ainsi que ses différentes données associées, mais cette instance n'a pas encore rempli les conditions nécessaires pour commencer son exécution.
- en cours d'exécution : l'instance du processus est en cours d'exécution et un ou plusieurs activités de cette instance de processus peuvent commencer, si les conditions liées à ces activités sont remplies.
- actif : une ou plusieurs activités de cette instance de processus ont commencé (un ou plusieurs éléments de travail ont été créés et reliés à l'instance d'activité appropriée) et des instances d'activités existent (dans certaines implémentations, des sous-états d'activités actives peuvent être pris en charge).
- suspendu : l'instance du processus existe toujours mais plus aucune activité ne peut être commencée jusqu'à ce que l'instance ne soit reprise (à nouveau active).
- achevé : les conditions de terminaison de l'instance du processus sont remplies et des opérations (liées au système) liées à la terminaison, comme la création d'informations d'audit, sont effectuées.
- tué : l'exécution s'est terminée (abandonnée anormalement) à cause d'une erreur ou d'une demande d'un participant.

- archivé : l'instance du processus (en général de longue durée) a été archivée en vue d'être reprise plus tard.

Les processus peuvent être suspendus, mais les activités peuvent être ne pas être interrompues, c'est-à-dire que une fois qu'une activité est commencée, il est impossible d'interrompre (suspendre puis reprendre, ou tuer) cette activité. Cela signifie que les fonctions servant à suspendre ou tuer une instance de processus ne peuvent prendre effet que si toutes les instances d'activités sont achevées et que le processus est retourné à son état "en cours d'exécution".

Passons maintenant aux instances d'activité et leurs états internes.

Définition - Instance d'une activité Une instance d'activité (appelé aussi instance d'étape, instance de noeud, instance de tâche, instance d'élément de travail) est la représentation d'une activité dans le contexte de l'exécution d'un processus, c'est-à-dire d'une instance de processus. Ses caractéristiques sont les suivantes :

- Une instance d'activité est créée, gérée et terminée par un WFMS dans le cadre d'une instance de processus, et ce conformément à la définition du processus.
- Chaque instance d'activité représente une invocation unique d'une activité, elle correspond à une seule instance de processus et utilise les données associées à cette instance de processus. Plusieurs instances d'activités peuvent être associées à une instance de processus et des activités s'exécutant en parallèle peuvent exister, mais une instance d'activité ne peut être associée à plus d'une instance de processus.
- Chaque instance d'activité peut normalement fournir des informations de contrôle, d'audit et d'état interne.

Définition - Etat d'une activité L'état d'une activité est la représentation de (conditions internes définissant) l'état d'une instance d'une activité à un instant bien déterminé. Les WFMS considèrent ces informations comme faisant partie des données de contrôle du workflow.

L'état d'une instance de processus en cours d'exécution est maintenu par le WFMS, mais certains systèmes étendent ces informations aux activités créées. Les WFMS ont différentes façons de représenter les états d'une activité et ont leur propre ensemble de définitions.

Le WF reference model identifie un ensemble d'états 3.4 que peut prendre une instance d'activité. C'est un modèle, bien sûr, donc une implémentation peut ne pas se limiter à ces états ou utiliser une autre représentation des états d'activité, ou ne pas en avoir du tout. La WfMC ne standardise pas le comportement interne des WFMS mais illustre les concepts de base nécessaires pour utiliser les API que la WfMC développe.

- inactif : l'instance d'activité a été créée mais n'est pas encore activée (elle ne remplit pas les conditions requises), aucun élément de travail n'existe pour cette activité.

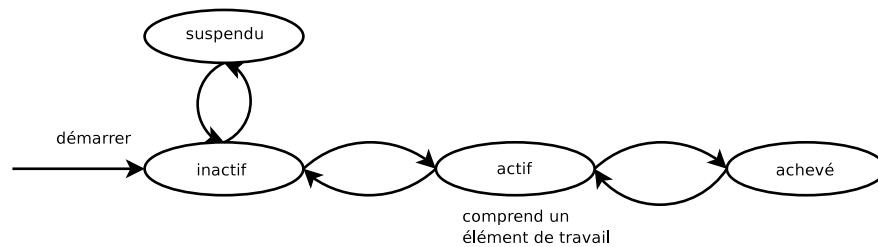


FIG. 3.4 – Les différents états qu'une activité peut avoir.

- **actif** : un ou plusieurs éléments de travail ont été créés pour cette instance d'activité et affectés à une instance d'activité (et à un participant) pour être traité.
- **suspendu** : l'instance d'activité existe toujours mais plus aucun élément de travail ne peut être entamé jusqu'à ce que l'instance d'activité soit reprise (certaines activités ne peuvent être suspendues).
- **achevé** : l'instance d'activité a atteint ses conditions de terminaison et des activités d'audit liées à la terminaison sont en cours.

3.3.4 Le routage des activités

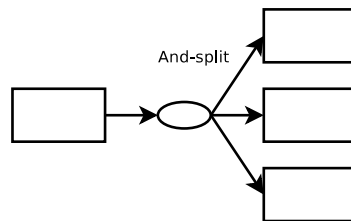
Une autre responsabilité du moteur de workflow est de gérer la succession des activités lors de l'exécution d'un processus. La définition du moteur de workflow parlait de routage parallèle et routage séquentiel, cette section examine et définit ces notions plus en détail.

Les deux premières définitions concernent l'exécution d'un processus. Selon la définition de ce processus, les activités peuvent s'exécuter en parallèle ou l'une après l'autre.

Définition - Routage parallèle Le routage parallèle (on parle aussi d'exécution parallèle de workflow ou exécution concurrente) est un morceau de l'instance d'un processus en cours d'exécution, dans lequel au moins deux instances d'activités sont en exécution parallèle dans le workflow, donnant lieu à plusieurs threads de contrôle. Il commence par un And-split et se termine par un And-join.

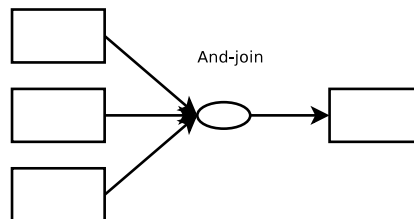
Définition - Routage séquentiel Le routage séquentiel est un segment de l'instance d'un processus en cours d'exécution dans lequel un certain nombre d'activités sont exécutées l'une après l'autre dans le même thread d'exécution (pas de And-split et And-join pendant un routage séquentiel). Le routage parallèle implique un point dans le workflow où le thread d'exécution se découple en deux threads d'exécutions, et un autre point où deux threads d'exécutions se rejoignent : les And-split et Or-split, les And-join et Or-join.

Définition - And-split Le And-split (figure 3.5) est un point de séparation dans le workflow par lequel un thread unique se divise en au moins deux autres threads qui sont exécutés en parallèle dans le workflow, permettant à plusieurs activités d'être exécutées en même temps (routage parallèle). Dès qu'un And-split est créé, il sépare le thread de contrôle de l'instance du processus. Ces threads s'exécuteront de manière autonome et indépendante jusqu'à l'arrivée d'un And-join.

FIG. 3.5 – *And-split*.

Dans certains WFMS, tous les threads créés lors d'un And-split commun doivent converger à un And-join commun. Dans d'autres systèmes, cette convergence peut se faire pour une partie de ces threads seulement à différents points And-join, entraînant potentiellement la création d'autres threads provenant d'autres points And-split.

Définition - And-join Un And-join (ou encore join, rendez-vous ou point de synchronisation, voir figure 3.6) est un point dans le workflow où au moins deux activités en exécution parallèle (routage parallèle) convergent en un thread unique. Le And-split peut être modélisé comme une pré-condition au And-join.

FIG. 3.6 – *And-join*.

Définition - Or-split Un or-split (on parle aussi de branchage conditionnel, routage conditionnel, switch ou branch) est un point dans le workflow où un thread unique prend une décision sur quelle branche prendre entre plusieurs branches de workflow alternatives. Un or-split est conditionnel et la transition (unique) vers l'activité suivante

est déterminée conformément au résultat de la (les) condition(s) de transition.

Définition - Or-join Le or-join (on dit également join ou de join asynchrone) est un point dans un workflow où au moins deux branches d'activités alternatives convergent vers une activité commune qui constitue l'étape suivant du workflow. Il n'y a pas de synchronisation requise car il n'y a pas ici d'exécution parallèle. Un thread de contrôle peut arriver d'une activité appartenant à n'importe quelle des alternatives.

Définition - Itération Une itération (synonymes : loop, while) est un cycle d'activités comportant l'exécution répétitive d'une ou plusieurs activités jusqu'à ce qu'une condition soit remplie.

Pour plus de clarté, il est bon de séparer les activités de routage (branchement conditionnel, And-split) des activités comprenant l'intervention d'un participant.

3.3.5 L'enchaînement de deux instances d'activités

Une des responsabilités du moteur de workflow est de gérer les changements d'état interne des instances de processus et d'activités. Lorsqu'une activité est achevée, son état interne est changé (vers l'état "achevé") ainsi que celui de l'instance du processus (état "en cours d'exécution") par le moteur de workflow et la prochaine transition est sélectionnée (il peut éventuellement y en avoir plusieurs). La transition déterminera l'activité suivante à instancier.

Définition - Pré-condition Une pré-condition (critère d'entrée, règle de début d'activité) est une expression logique qui peut être évaluée par un moteur de workflow pour décider si une instance de processus ou d'activité au sein d'une instance de processus peut être lancée. Par ailleurs, notons que :

- Une ou plusieurs pré-conditions peuvent être définies comme critère d'entrée (de démarrage) pour une instance d'activité ou de processus particulier.
- L'expression d'une pré-condition peut se référer à des données liées au workflow, des données liées à des variables système comme une date et des événements extérieurs.
- Les pré-conditions sont définies dans la définition du processus.
- Un And-join peut être défini en termes de pré-condition, qui requiert que chacun des threads convergeant se trouve dans un état spécifique.

Définition - Post-condition Une post-condition (critère de sortie, règles de terminaison d'activité) est une expression logique qui peut être évaluée par un moteur de workflow pour décider si une instance de processus ou d'activité au sein d'une instance de processus peut être (considérée comme) achevée. Voici quelques remarques concernant les post-conditions :

- Une ou plusieurs post-conditions peuvent être définies pour spécifier un critère de terminaison pour une instance d'activité ou de processus particulier. De telles conditions peuvent comprendre une itération dans laquelle l'exécution d'une ou

plusieurs activités est répétée jusqu'à ce que la (les) post-condition(s) soient remplies.

- L'expression d'une post-condition peut se référer à des données relatives au workflow, des données liées à des variables système comme une date et des événements extérieurs.
- Les post-conditions sont définies dans la définition du processus.

En pratique, les pré-conditions reposent notamment sur des And-join : certaines activités doivent être terminées pour que les pré-conditions de l'activité suivante soient remplies. Il existe aussi d'autres types de pré-condition : les pré-conditions sur des valeurs tirées des données relatives au workflow. Elles sont exprimées non pas sous forme de condition de démarrage d'une activité (dans ce cas l'activité est créée et est mise en état d'inactivité, donc n'est pas démarrée et n'a pas d'élément de travail associé) mais plutôt sous forme de condition de transition entre 2 activités (dans ce cas, si la pré-condition n'est pas remplie, l'activité n'est même pas créée). Ce qui nous amène aux définitions suivantes.

Définition - Transition (entre deux activités) Une transition est un moment dans l'exécution de l'instance d'un processus pendant lequel une activité se termine et le thread de contrôle passe à une autre activité dont l'instance est créée (si ce n'est pas déjà fait) et que le moteur démarre (si les pré-conditions sont remplies). Voici quelques remarques concernant les transitions entre deux activités :

- Voir aussi : transition de changement d'état, condition de transition.
- Une transition peut être non conditionnelle : la terminaison d'une activité mène toujours au démarrage d'une autre activité.
- Elle peut être conditionnelle : la séquence des opérations dépend d'une ou plusieurs conditions de transition.
- Les transitions et leurs conditions sont définies dans la définition du processus.

Définition - Condition de transition (entre 2 activités) La condition de transition (ou encore règle de transition, condition de routage, règle de processus, règle de processus organisationnel) est une expression logique qui peut être évaluée par le moteur de workflow pour décider l'exécution de la séquence d'activité au sein d'un workflow. Voici quelques remarques concernant ces conditions de transition :

- Une ou plusieurs conditions de transitions peuvent être définies pour qu'une évaluation ait lieu au commencement ou à la terminaison de l'exécution d'une activité, ou suite à un événement externe.
- L'expression des conditions de transition peuvent se référer aux données du workflow et aux variables système comme la date.
- Les conditions de transition caractérisent le flux entre les activités et sont utilisées pour obtenir l'exécution d'une séquence particulière d'activités, parallèle ou séquentielle.
- Les conditions de transitions sont définies dans la définition du processus.
- Certains WFMS ne définissent pas explicitement les conditions de transition mais utilisent une combinaison de pré- et post-conditions pour obtenir le même effet.

3.3.6 Les données dans le WFMS

D'après sa définition, un des rôles du moteur de workflow est de maintenir des informations sur les instances de processus et d'activité. Ces informations sont des données sur leur état interne ou sur les données manipulées lors de l'exécution des activités. Cette section définit les catégories de données intervenant dans un système de gestion de workflow.

Dans un WFMS, il y a 3 catégories de données (figure 3.7) les données de contrôle du workflow, les données relatives au workflow, et les données des applications liées à une activité dans le workflow.

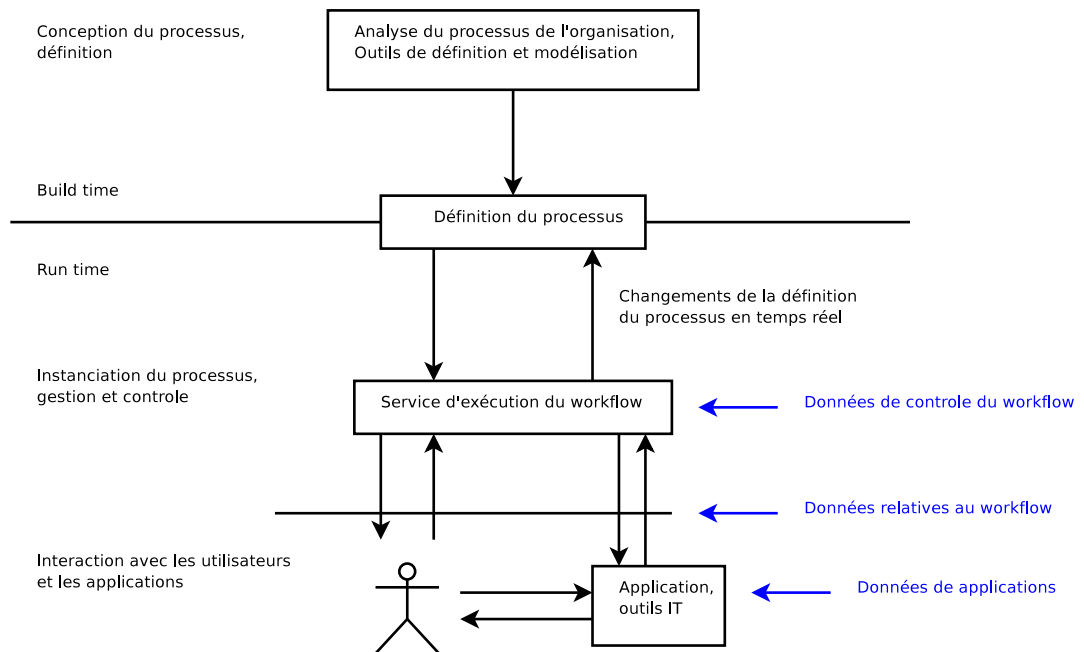


FIG. 3.7 – Les données dans le WFMS.

3.3.6.1 Les données de contrôle du workflow

La première catégorie de donnée est celle des données de contrôle du workflow. Le service d'exécution de workflow contrôle des moteurs de workflow qui maintiennent des données de contrôle internes pour identifier l'état des instances de processus et d'activité et d'autres informations internes. Ces données ne sont pas accessibles ni échangeables telles qu'elles par la WAPI (Workflow Application Programming Interface, l'interface entre les différents composants d'un WFMS) mais certains contenus d'information peuvent être transmis en réponse à certaines commandes (requête de l'état d'un processus, obtenir des métriques, etc) sous un certain format.

Définition - Données de contrôle du workflow Les données de contrôle du workflow (on parle aussi de données sur l'état du moteur de workflow, données sur l'état du service d'exécution du workflow) sont des données gérées par le moteur de workflow uniquement. Elles ne sont pas accessibles par les autres applications. Notons que :

- Les données de contrôle du workflow représentent l'état dynamique du WFMS et ses instances de processus et d'activité.
- Ces données peuvent être maintenues de façon persistantes pour récupérer l'état d'exécution avant l'apparition d'une erreur ou pour fournir des informations d'audit.
- Quelques exemples de données de contrôle :
 - information sur l'état d'un processus particulier
 - information d'état d'une activité (en cours ou non)
 - information sur la récupération et les points de redémarrage du processus

3.3.6.2 Les données relatives au workflow

La deuxième catégorie de données est celle des données relatives au workflow (workflow relevant data), ces données sont utilisées par le WFMS pour déterminer les conditions de transition d'état dans le workflow, autrement dit elles concernent le choix de l'activité suivante qui sera exécutée (autrement dit les pré-conditions de création d'une instance d'activité). Elles sont utilisées aussi lors de l'exécution des activités et reflètent l'état d'avancement du travail.

De telles données sont donc potentiellement accessibles par les applications (invoquées comme support informatique pour effectuer l'une ou l'autre activité du processus) et seront donc transférées entre les activités par un des moteurs du service d'exécution de workflow. Dans un environnement hétérogène comprenant plusieurs moteurs de workflow et où l'exécution peut s'étendre sur plusieurs de ces moteurs, ces informations doivent être transférées entre les moteurs (ou partagées par tous ces moteurs) et le processus peut nécessiter des correspondances entre les noms utilisés et une conversion des données.

Définition - Données relatives au workflow Les données relatives au workflow (ou encore données liées au flux du processus) sont des données utilisées par le WFMS pour déterminer les transitions entre deux activités d'une instance de processus, par exemple les conditions de transition (entre deux activités) et l'affectation à un participant (une participation humaine est obligatoire pour certaines activités, un participant manquant empêche l'activité de s'exécuter tout comme non conformité d'une pré-condition, ce qui fait du participant une sorte de "donnée relative au workflow"). De plus :

- Les données relatives au workflow peuvent être manipulées par le moteur de workflow et par les applications de workflow.
- Elles peuvent être accessibles aux activités postérieures ou à d'autres instances de processus et donc affecter le choix de l'activité suivante à exécuter.

3.3.6.3 Les données des applications de workflow

La troisième catégorie de données est celle des données d'applications de workflow. Une application de workflow est utilisée lors des activités par des participants comme support pour effectuer leur travail. Une application de workflow peut-être aussi un programme autonome, fonctionnant sans intervention d'un participant. Les applications de workflow accèdent aux données relatives au workflow, mais ont aussi leurs propres données "privées", que le reste du système (notamment le moteur de workflow) ignore.

La manipulation des données des applications a lieu lors des différentes activités d'un processus. Ces données sont modifiées directement par l'application elle-même (si elle est autonome, elle fonctionne sans participant) ou par le participant, en interagissant avec l'application. Les données des applications ne sont pas utilisées par le service d'exécution de workflow et ne concernent que les applications et les tâches des participants exécutées lors de l'activité.

Définition - Données des applications Les données des applications sont des données spécifiquement liées aux applications de workflow et ne sont pas accessibles par le WFMS.

Remarques :

- Ces données sont gérées strictement par des applications de workflow supportant les instances de processus. Celles qui déterminent des changement d'état du workflow sont des données relatives au workflow.
- Les applications sont soit des applications clientes (interface 2 du workflow reference model) soit des applications invoquées (interfaces 3 du workflow reference model).

La relation entre les applications et les données pertinentes ou les données des applications est normalement définie dans la définition du processus. Il peut y avoir échange direct ou indirect de données. L'échange indirect se fait grâce à un espace partagé d'objets et comprend une relation explicite définissant le nom d'un objet et son accès pour les applications. L'échange direct de données déplace physiquement les données lors de la fin d'une activité pour quelles soient accessibles à la nouvelle activité.

3.3.7 L'échange de données (direct - indirect)

Les données relatives au workflow sont accédées et modifiées par le moteur de workflow (pour tester les conditions de transition, par exemple) et par les applications que les participants utilisent lors des activités (c'est-à-dire les applications invoquées). Les composants d'un WFMS doivent donc s'échanger des données.

L'échange de données relatives au workflow peut se faire de deux manières : l'échange direct et l'échange indirect. L'échange direct pourrait se faire en utilisant la WAPI (Workflow Application Programming Interface, une interface mise au point pour que les composants d'un WFMS puissent communiquer avec le composant principal, qui est le système d'exécution de workflow ou avec d'autres WFMS), les données seraient

transmises dans ce cas en paramètres des méthodes de l'interface. L'échange indirect consiste en un répertoire de données partagé entre les composants.

La WAPI consiste en cinq interface, dont le détail sera expliqué dans la section suivante consacrée à la WAPI. Voici quelques considérations concernant l'échange de données directes et indirectes, entre les différents composants :

- Les applications clientes, comme le gestionnaire de liste de processus (interface 2 de la WAPI) : par échange direct, les données peuvent être incorporées dans les éléments de travail et extraites depuis la liste de travail pour être présentée à l'utilisateur. Par échange indirect, les données sont stockées dans un répertoire partagé et sont transmises par référence (incorporée dans l'élément de travail) aux utilisateurs et aux applications.
- Les applications invoquées (interface 3 de la WAPI) : l'échange des données relatives au workflow va dépendre de la nature de l'interface d'invocation d'application (à ce stade, la WAPI est très vague en ce qui concerne l'invocation d'applications) et peut requérir du service d'invocation d'incorporer les données dans un protocole spécifique.

L'interopérabilité entre moteur de workflow (interface 4 de la WAPI) : de même que pour les applications clientes, mais différents systèmes peuvent supporter différentes approches d'échange de données d'application, l'utilisation d'un composant de traduction entre les 2 parties peut être nécessaire.

3.4 La WAPI (*Workflow Application Programming Interface*)

La WfMC (*Workflow Management Coalition*), en parallèle avec le modèle de référence des systèmes de gestion de workflow, développe une interface pour que les composants définis dans le modèle de référence (les applications invoquées, les gestionnaires de liste de travaux, les outils d'administration) puissent communiquer avec le composant principal, c'est-à-dire le service d'exécution de workflow. La *Workflow API* [14] comporte 5 interfaces.

3.4.1 Interface 1 : l'importation et l'exportation de définition de processus

3.4.1.1 Les outils de définition de processus

Différents outils peuvent être utilisés pour analyser, modéliser et décrire un processus organisationnel. Le modèle de référence de workflow ne se préoccupe pas de ces outils. De tels outils peuvent donc être incorporé au système ou être séparé.

Si les outils sont incorporés au système, les définitions de processus produites par ces outils seront utilisées par le domaine du workflow et pourront être utilisées par des programmes via des API pour écrire et retirer les informations qu'elle contiennent. Ces définitions sont stockées dans un répertoire pour que d'autres outils et d'autres systèmes puissent les récupérer. Chaque produit de workflow a son propre langage de

définition de workflow et c'est l'interface d'importation et d'exportation de définition de processus qui permet une flexibilité dans un domaine de workflow hétérogène.

L'analyse et la modélisation de processus peut définir un processus dans un contexte d'une structure organisationnelle. Un modèle de l'organisation sera donc incorporé dans ces définitions de processus, qui comprendra des objets liées à cette structure organisationnelle comme des rôles, des relations entre les participants.

3.4.1.2 L'échange de définitions de workflows

L'interface entre les outils de définitions de processus et le logiciel de gestion de l'exécution des workflows est appelé l'interface d'importation et d'exportation de définition de processus (figure 3.8). Cette interface consiste en un format d'échange de définition de processus et un ensemble d'API permettant l'échange d'informations contenues dans une définition (l'intégralité ou une partie des informations, comme un attribut d'une activité particulière).

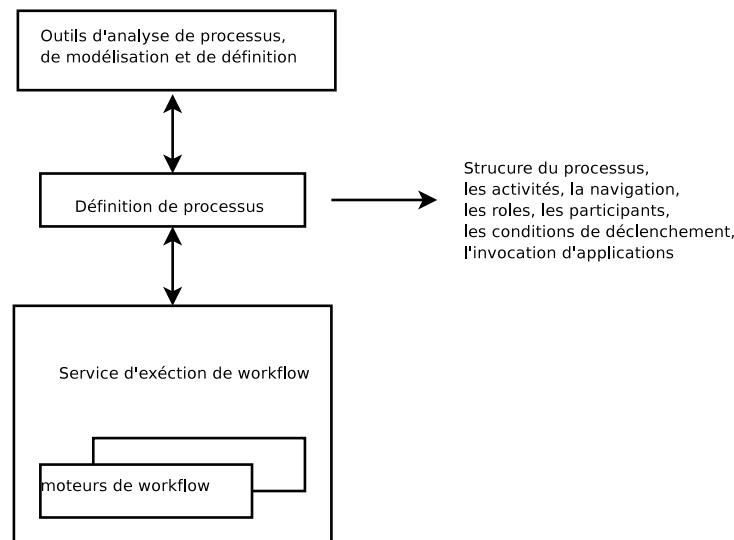


FIG. 3.8 – *L'interface 1 : importation et exportation de définition de processus.*

Une forme standardisée pour ces définitions de processus est utile pour plusieurs raisons :

1. Elle permet une séparation, l'indépendance entre les environnements de build-time et de run-time, permettant à la définition d'être générée par un outil spécialisé et d'être utilisée comme input pour différents systèmes de workflow.
2. Elle permet à la définition d'être exportée vers d'autres systèmes de workflow pour qu'ils puissent coopérer et former un environnement d'exécution distribué.

La WfMC s'est intéressé à ces domaines :

1. Un méta-modèle décrivant les objets (appelés aussi entités), ainsi que leurs relations et leurs attributs, que l'on peut trouver dans une définition de processus,

ce méta-modèle sert de base aux formats d'échange de définition de processus entre différents produits de workflow.

2. Les appels API entre systèmes de workflow et entre un système de workflow et un outil de définition de processus, fournissant une manière commune d'accéder aux informations contenues dans ces définitions de processus.

3.4.1.3 Méta-modèle de base pour les définitions de processus

La WfMC a développé un modèle pour les définitions de processus (figure), identifiant les objets de base permettant un premier niveau d'échange de définition de processus simples. D'autres objets peuvent être ajoutés selon les implémentations et les fonctionnalités.

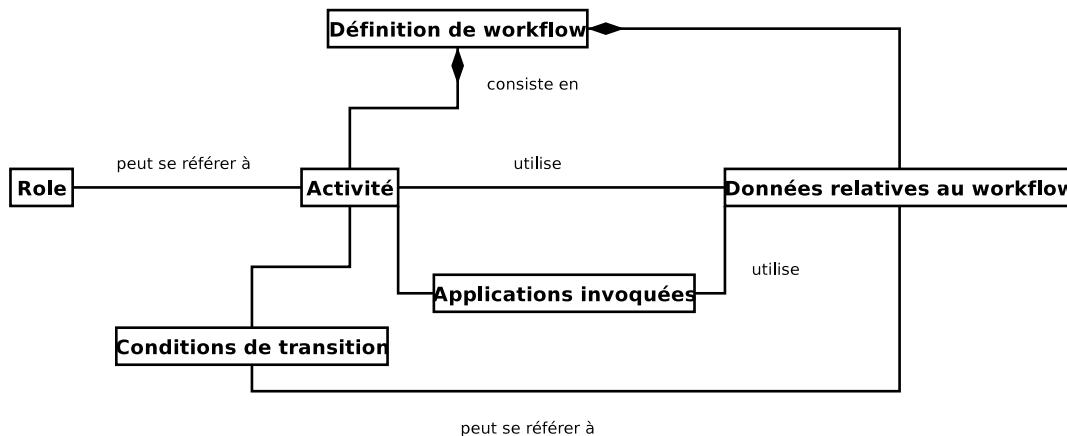


FIG. 3.9 – Le méta-modèle d'une définition de processus.

Voici quelques attributs qui peuvent être envisagés pour ces objets :

- pour l'objet définition de workflow : le nom et la version du workflow, les conditions de commencement et de terminaison, les données d'audit, de sécurité et de contrôle.
- pour l'objet activité : son nom, son type (sous-workflow, activité atomique, ou autre), les pré-conditions et post-conditions de l'activité, d'autres contraintes de planification.
- pour l'objet conditions de transition : les expressions conditionnelles.
- pour les données relatives au workflow : le nom et le chemin d'accès des données.
- pour l'objet rôle : le nom du rôle.
- pour l'objet applications invoquées : le nom, ses paramètres, le chemin d'accès à cette application.

3.4.1.4 L'API pour accéder aux définitions de processus

La WfMC est en train de développer des API pour accéder aux données contenues dans les définitions de processus. Elles reprennent la connexion d'un participant au système, des opérations permettant d'obtenir la liste des définitions de processus, et des méthodes d'accès aux différents objets et attributs d'une définitions de processus.

Par ailleurs, la WfMC a utilisé XML pour définir un langage de référence pour les définitions de processus : XPDL (XML Process Definition Language). Ce langage qui est développé actuellement en parallèle avec un autre standard, le BPMN (Business Process Modeling Notation), un langage de définition de processus qui s'exprime à travers des diagrammes de flux. Le chapitre suivant montre comment un même processus peut-être exprimé en XPDL et BPMN, et montre les workflow patterns présents dans les deux langages.

3.4.2 Interface 2 : l'interface avec les applications clientes

3.4.2.1 Les applications de workflow

Une application de workflow, appelée aussi application d'un WFMS, est un terme général utilisé pour dénoter un logiciel qui interagit avec un moteur d'exécution de workflow via le service d'exécution du workflow (workflow enactment service). Ce logiciel se charge des traitements nécessaires pour supporter une ou plusieurs activités.

Le WF reference model classe ces application en deux groupes :

1. Les applications clientes, qui requièrent des services à un moteur de workflow.
2. Les applications invoquées, qui supportent certaines activités ou éléments de travail, et sont appelées par le WFMS.

Définition - Application cliente Une application cliente est une application qui interagit avec un moteur de workflow, lui demandant des services. Les applications clientes peuvent interagir avec le moteur de workflow pour différentes raisons :

- La gestion de la liste de travaux
- La création d'une instance de processus et d'autres fonctions liées (arrêter, reprendre)
- La récupération (et la manipulation) des données de la définition de processus (outil de modélisation de processus)
- Des fonctions d'administrations (par exemple suspendre l'utilisation de certaines définitions de processus)

Le WF reference model reprend une interface pour l'interaction avec ces applications clientes, qui comprennent des API pour ces fonctions.

3.4.2.2 Le gestionnaire de liste de travaux

Une certaine application cliente d'un WFMS, appelée "gestionnaire de liste de travaux" (worklist handler), est un logiciel interagissant avec un utilisateur final pour

les activités qui nécessitent une participation humaine. Ce logiciel peut faire partie intégrante du produit de management de workflow ou être un logiciel séparé, écrit spécialement pour un utilisateur particulier dans le but de le satisfaire au mieux. Il est donc nécessaire d'avoir un mécanisme de communication flexible entre le service d'exécution de workflow et les applications clientes pour supporter les différents scénarios.

L'interaction entre l'application cliente et le service d'exécution de workflows se fait donc à travers une interface bien définie qui repose sur les concepts :

- de liste de travaux,
- d'éléments de travail affectés à un participant particulier.

Voici une définition pour ces nouveaux concepts.

Définition - Élément de travail Un élément de travail est la représentation du travail qui doit être effectué par un participant dans le contexte d'une activité d'une instance de processus. Voici quelques synonymes : travail (révision d'un document, remplissage d'un formulaire,...), objet de travail, élément d'une liste de travail, tâche.

On peut ajouter sur les éléments de travail ces quelques remarques :

- Une activité est typiquement composée de un ou plusieurs éléments de travail qui constituent la tâche que doit effectuer le participant dans le cadre de son activité. Dans certains cas, une activité est complètement prise en charge par une application sans le concours d'un participant. Il n'y a donc parfois pas besoin d'affecter des éléments de travaux pour un activité.
- Les éléments de l'activité sont présentées au participant via sa liste de travaux, qui contient les détail des éléments de travail affectés au participant. Le gestionnaire de la liste de travaux interagit directement avec la liste à la place (et à la demande) du participant.
- Le contrôle et la progression des éléments de travail repose sur le participant et le gestionnaire de liste de travaux, pas sur le WFMS. Le WFMS est prévenu de l'état des éléments (en cours ou terminé) par interface avec le gestionnaire de liste de travaux (voir WAPI interface).
- Des outils et applications peuvent être invoquées pendant le traitement de l'élément, ou indépendamment du WFMS, qui est simplement prévenu de la fin d'un élément particulier.

Définition - Liste de travaux La liste des travaux (liste d'attente de travaux, liste d'entrée, to-do list) est une liste d'éléments d'activités associée à un participant (ou plusieurs participants partageant la même liste de travaux). La liste de travaux fait partie de l'interface entre un moteur de workflow et un gestionnaire de liste de travaux. Il faut savoir de plus que :

- En général, un gestionnaire de liste de travaux fera une requête au moteur de workflow pour obtenir des éléments d'activité pour créer une liste de travaux.
- Dans certain WFMS, des éléments d'activités sont placés directement dans les listes de travaux par le moteur de workflow pour être accédé et traité par après par le gestionnaire de liste de travaux.

Définition - Gestionnaire de liste de travaux Le gestionnaire de liste de travaux (WFM front end, WFM application, application de la to-do list, manager des tâches) est un composant logiciel qui gère l'interaction entre le participant (ou un groupe de participant) et la liste de travaux maintenue par le moteur de workflow. Il permet aux éléments de travail de passer du WFMS jusqu'au participants et permet aux notifications de terminaison de travaux et autres informations sur l'état du travail de passer des participants jusqu'au WFMS. Ajoutons ces quelques remarques :

- Le gestionnaire de liste de travaux (figure 3.10) peut être un produit fourni avec le WFMS ou une application autonome. Il peut communiquer avec plusieurs WFMS, réunissant tous les éléments de travail en une seule liste et la présenter au participant.
- Des fonctions liées au gestionnaire de liste de travaux sont :
 - sélectionner un élément de travail
 - réaffecter un élément de travail
 - aviser le WFMS de la terminaison d'un élément de travail
 - invocation d'un outil ou d'une application lors du traitement de l'élément de travail
- L'interface WAPI du WfMC comprend des appels API standards pour la communication entre le gestionnaire de la liste des travaux et le moteur de workflow.

Les fonctions de base de l'interface entre le moteur de workflow et le gestionnaire de liste de travaux assurent que le moteur de workflow peut accéder à la liste de travaux d'un participant pour lui affecter des travaux, le gestionnaire de liste de travaux puisse récupérer les éléments de travaux et les présenter au participant. Cela laisse plusieurs possibilités d'implémentation (figure 3.11).

Il y a quatre possibilités d'implémentation :

1. le système sur un seul hôte : le gestionnaire de travaux est une application locale, il peut accéder à la liste de travaux du participant par appel local au service d'exécution de workflow.
2. les composants accèdent à une base de données partagée : les composants se transmettent l'information indirectement en utilisant un serveur de données comme intermédiaire.
3. envoi par courrier électronique : le moteur de workflow envoie un courrier électronique au participant, l'application cliente comprend donc un service de courrier électronique.
4. Par appels distants : le gestionnaire accède aux listes de travaux par un appel distant à l'interface du système d'exécution de workflow.

L'activation d'un élément de travail (lancer l'application et s'occuper des données liées au workflow) peut être contrôlée par l'application cliente ou par le participant. Une série de procédures sont définies entre l'application cliente et le service d'exécution de workflow pour permettre d'ajouter des éléments à la liste de travaux, de retirer ces éléments une fois achevés, de suspendre certaines activités. Le nom et l'adresse de l'application est fournie au gestionnaire liste de travaux par le service d'exécution de workflow, des fonctions dans l'interface 3 (sur les applications invoquées) seront prévues à cet effet si nécessaire. Potentiellement, un gestionnaire de liste de travaux

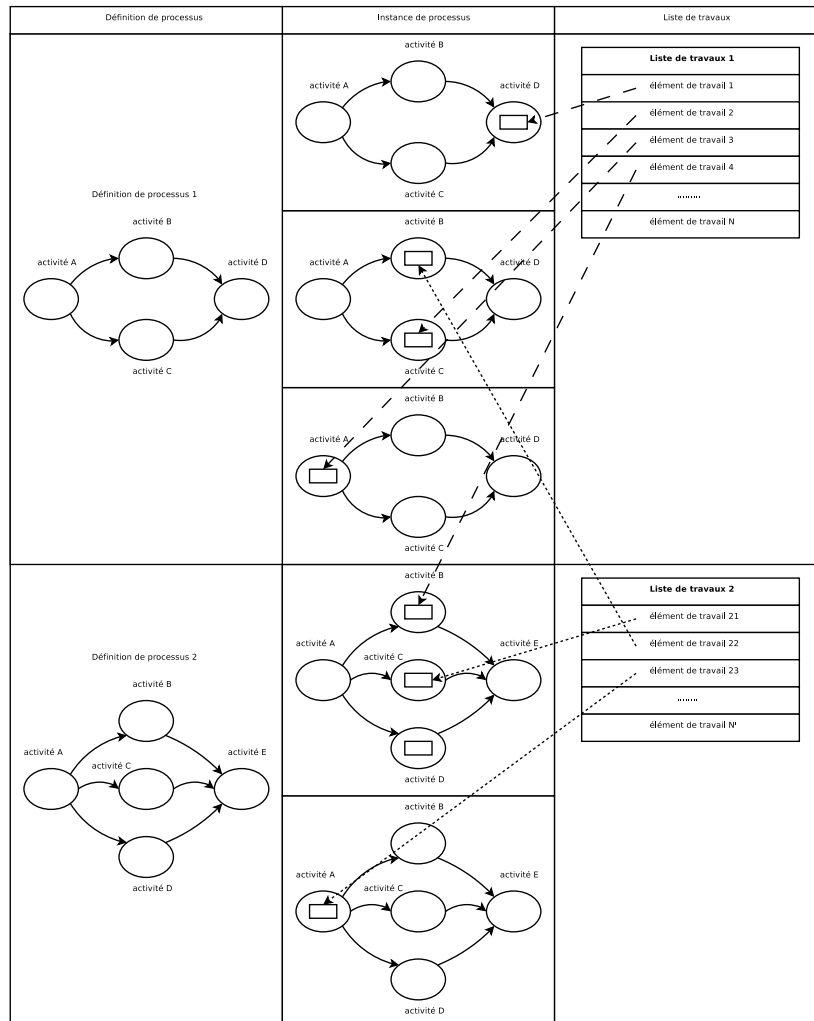


FIG. 3.10 – Illustration d'un gestionnaire de liste de travaux.

peut gérer les éléments de travaux de différentes instances d'un même processus ou de processus différents, il peut interagir avec différents moteurs de workflow et services d'exécution de workflow. L'interface entre les applications clientes et le service d'exécution de workflow doit donc être suffisamment flexible pour prendre en compte les identificateurs d'un processus ou d'une activité, les noms et adresses des ressources, les références vers des données, les différentes structures de données, les différentes possibilités de communication.

3.4.2.3 La communication entre les applications clientes et le service d'exécution de workflow

La WAPI (3.12) tente de répondre à ces exigences. Elle fournit le moyen d'accéder au service d'exécution de workflow et à la liste de travaux depuis le gestionnaire de

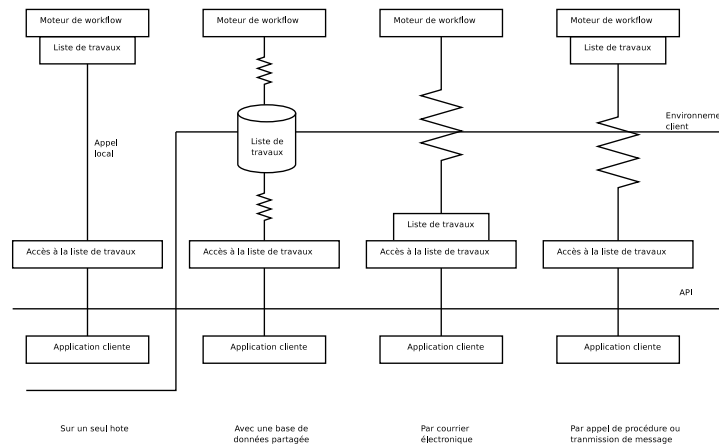


FIG. 3.11 – *Implémentations possibles d'un gestionnaire de liste de travaux.*

liste de travaux. Les API et les paramètres devront être adaptés pour le moyen de communication adopté par une implémentation particulière.

3.4.2.4 Les API pour les applications clientes

Les commandes sont des opérations permettant de travailler sur des instances particulières de processus et d'activités individuelles, sur un ensemble d'instances de processus et d'activités et sur les listes de travaux.

- L'établissement d'une session : connexion et déconnexion entre les systèmes participants.
- Les opération concernant les définitions de workflow : fonctions d'accès aux définitions de processus.
- Les fonctions de contrôle de processus : création, démarrage, interruption, redémarrage et terminaison... changements d'état d'instances de processus ou d'activité, accès ou modification d'un attribut de cette instance.
- Les fonctions d'état (statut) d'un processus
- Les fonctions liées aux liste de travaux
- Les fonctions de supervision de processus
- Les fonctions de gestion de données : accès ou remplissage de données relatives au workflow ou données des applications
- Les fonctions d'administration
- Les fonctions d'invocation d'applications : une base fonctionnelle qui permet au gestionnaire de liste de travaux d'invoquer des applications, ç-à-d permettre l'accès aux attributs des processus, activités et éléments de travaux.

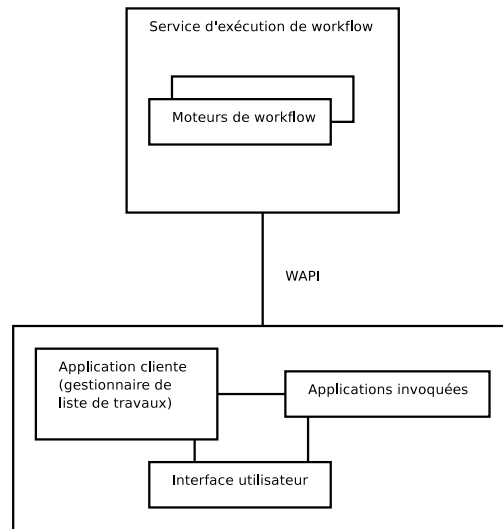


FIG. 3.12 – L'interface avec une application cliente.

3.4.3 Interface 3 : l'interface avec les applications invoquées

3.4.3.1 Les applications invoquées

Une application invoquée est une sorte particulière d'application de WFMS. Elles sont utilisées par les participants lors de l'exécution d'une instance de processus, afin de réaliser leur travail.

Définition - Application invoquée Une application invoquée est une application de workflow invoquée par le WFMS pour automatiser une activité, entièrement ou partiellement, ou pour aider un participant à effectuer un élément de travail. Elle peut être une fonction d'un moteur de workflow ou du gestionnaire de liste de travaux, et peut être déclenchée directement par le WFMS ou indirectement par un "tool agent" qui fournit un mécanisme général d'invocation de manière indépendante à un WFMS particulier. Le WF reference model comprend une interface pour les fonctions d'invocation d'applications.

La WfMC prend comme hypothèse qu'aucune implémentation d'un système de workflow ne permet d'invoquer toutes les applications possibles qui pourraient exister dans un environnement hétérogène. Cela supposerait l'invocation d'application quelque soit la plateforme et le réseau utilisés, et un format et un encodage commun pour toutes les données relatives au workflow et aux applications, ou une transformation de ces données pour qu'elles puissent être prises en charge par un sous-système particulier.

Certaines implémentations utilisent des "application agents" ou "tool agents" qui gèrent toutes les méthodes d'invocation derrière une interface standard utilisée par

le service d'exécution de workflow. Une autre possibilité est de développer des applications spéciales pour le système de workflow, qui utiliseront un ensemble d'API standard pour communiquer avec le service d'exécution de workflow. Le travail de la WfMC dans ce domaine est de développer des types et des API standards pour une utilisation ultérieure, pour cacher la méthode d'invocation utilisée (appel d'une procédure local, un script shell, un appel à travers un ORB, un appel distant, un envoi de message, opération synchrones ou asynchrones...) derrière une interface standard.

3.4.3.2 La communication entre les applications invoquées et le service d'exécution de workflow

Dans le cas le plus simple, l'invocation d'application est gérée localement par rapport au moteur de workflow, utilisant les informations de la définition de processus pour identifier la nature de l'activité, l'application qui doit être invoquée et les données nécessaires. L'application invoquée peut être locale ou distante, l'adresse de l'application est reprise dans la définition du processus. Certaines applications seront conçues de manière à pouvoir directement interagir avec le système d'exécution de workflow, d'autres nécessiteront une interface intermédiaire (figure 3.13).

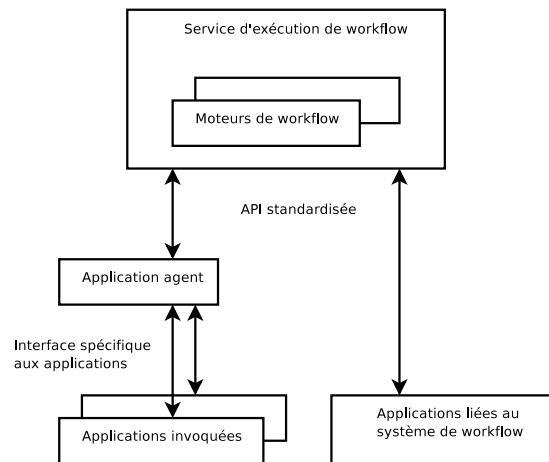


FIG. 3.13 – *L'interface avec une application invoquée.*

3.4.3.3 Les API pour l'invocation d'applications

Le détail quant aux API publiées par la WfMC est encore en cours de développement. Voici un début pour l'ensemble des commandes qui concernent l'invocation d'applications. Certaines sont

- L'établissement d'une session : connexion et déconnexion d'une application ou d'une application tool au moteur de workflow (ce n'est pas plutôt au service

d'exécution de workflow).

- Les fonctions de management d'une activité : commencer une activité, la suspendre, recommencer, abandonner (appel du moteur de workflow vers l'application); notification d'achèvement de l'activité, signaler un événement, récupérer les attributs d'une activité (de l'application vers le moteur de workflow).
- Les fonctions de gestion des données : transmettre les données relatives au workflow (pré-activité : du moteur de workflow vers l'application, post-activité : de l'application vers le moteur de workflow), transmettre les données de l'application ou l'adresse des données.

Des scénarios plus complexes nécessitent une communication entre des moteurs de workflow hétérogènes aussi bien après la phase de développement de la définition du workflow que pendant son exécution. Cet aspect est considéré par l'interface 4, l'interopérabilité de workflow.

3.4.4 Interface 4 : l'interopérabilité entre systèmes de workflows

3.4.4.1 Des services d'exécution de workflows hétérogènes

Un service d'exécution de workflow est un service comprenant un ou plusieurs moteurs de workflow, et qui permet de créer, gérer et exécuter des instances de workflow (processus) particulières. Les applications interagissent avec ce service via une partie de la WAPI. Un service d'exécution de workflow peut être composé de moteurs de workflow se présentant comme une unité homogène (c'est-à-dire formant un domaine de workflow) ou de moteurs de workflow formant un ensemble hétérogène. Nous avons donc un nouveau concept à définir : le domaine de workflow.

Définition - Domaine de workflow Un domaine de workflow est un service de gestion de workflow comprenant un ou plusieurs moteurs de workflow, gérés de manière à présenter une unité homogène, qui présentent les mêmes fonctions d'administration, c'est-à-dire le nommage commun des workflow (processus et activités), des participants, une interprétation commune des définitions de processus et des transitions d'états, un modèle d'organisation et de rôles communs, ...

Un domaine de workflow est construit à partir de produits homogènes. L'interface 4 du workflow reference model décrit l'interface entre le service d'exécution de workflow et des domaines de workflow, de manière à incorporer des produits hétérogènes. Un des objectifs de la WfMC est de définir des standards permettant à des systèmes de workflows proposés par des fournisseurs différents de se passer l'un à l'autre des éléments de travail.

Les différents produits sur le marché sont tous différents par nature. La WfMC ne veut pas forcer les fabricants à devoir choisir entre un produit centré sur les besoins d'un client particulier et un produit offrant l'interopérabilité avec d'autres produits différents. La WfMC a donc considéré d'abord plusieurs scénarios complexes d'interopérabilité dans lequel plusieurs moteurs de workflow différents coopèrent pour offrir un unique service d'exécution de workflow. Un tel but est loin d'être réalisé car pour cela il faudrait d'abord que tous ces moteurs puissent interpréter le même langage de description de processus et utilisent les mêmes données de contrôle de workflow,

pour partager une vue commune des états des processus s'exécutant à travers tous ces moteurs hétérogènes.

Un but plus réalisable serait de permettre à un moteur de workflow de transférer une partie d'un processus pour que son exécution soit assurée par un autre service d'exécution de workflow. Quatre modèles d'interopérabilité ont été identifiés par la WfMC, offrant différents niveaux de capacité (des niveaux de plus en plus hauts).

Scénario 1 : la chaîne La chaîne (voir figure 3.14), ou processus connectés de manière discrète, est un modèle qui permet une connexion entre un processus A et un processus B. Dans l'illustration, la connexion se fait de la fin du processus A vers début du processus B, mais les points de connexions peuvent se trouver à n'importe quel autre endroit.

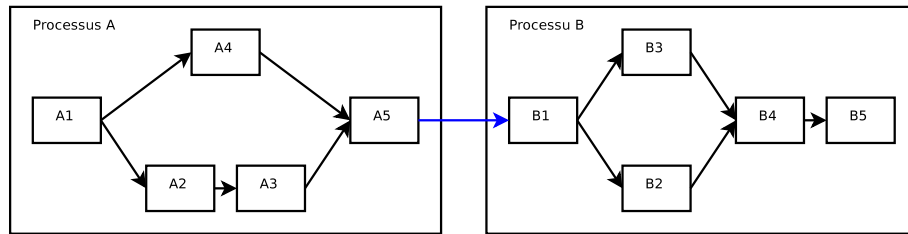


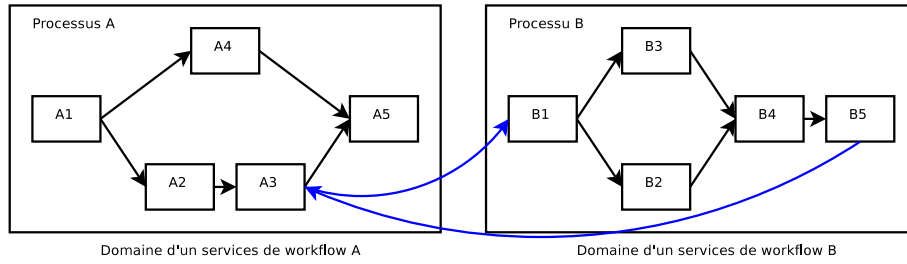
FIG. 3.14 – La chaîne.

Ce modèle supporte le transfert d'une instance de processus ou d'activité entre les deux environnements, cette instance va continuer à s'exécuter dans l'autre environnement sans autre contrainte (comme par exemple un point de synchronisation ultérieur entre les deux processus). Pour l'implémentation, cela implique une fonction assurant cette transition, en prenant compte la conversion du format des données, la correspondance entre les noms des processus et des activités, etc.

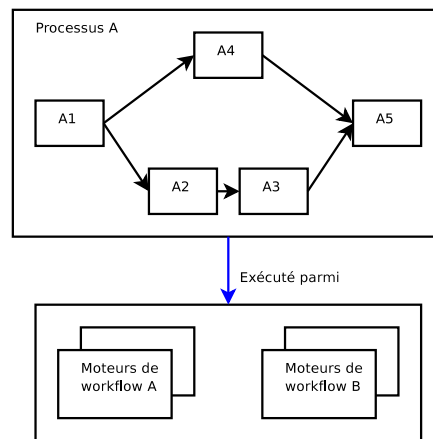
Scénario 2 : Les sous-processus encapsulés Le modèle des sous-processus encapsulés (figure 3.15) permet qu'un processus (plus précisément un sous-processus) exécuté dans un domaine de workflow particulier soit encapsulé dans une tâche (une activité) unique située dans un processus "hiérarchiquement supérieur" s'exécutant dans un domaine de workflow différent. Cette relation hiérarchique peut se prolonger en profondeur sur plusieurs niveaux, formant alors des sous-sous processus.

Dans la figure 3.15, le service (d'exécution) de workflow A comprend une activité A3 qui est exécutée en tant que processus à part entière B dans le service d'exécution de workflow B, qui rends le contrôle à A à la fin de ce processus.

Scénario 3 : Le point-à-point Le modèle point-à-point (ou processus connectés de manière non-discrète) permet à chacune des activités d'être exécutées par un service d'exécution de workflow particulier (composé de plusieurs moteurs) choisi parmi un ensemble de services d'exécution de workflow formant un domaine partagé. Les activités A1, A2 et A5 seront exécutées par le service A et les activités A3, A4 et A6

FIG. 3.15 – *Le sous-processus encapsulé.*

par le service B (figure 3.16). Le processus serait exécuté de manière transparente, les interactions nécessaires entre les différents moteurs de workflow ayant lieu sans intervention d'un utilisateur.

FIG. 3.16 – *Le point à point.*

Ce scénario nécessite que les différents services d'exécution de workflow aient un ensemble d'API communes et qu'ils puissent interpréter le même langage de définition de workflow, utilisé soit dès la définition du processus soit entre les différents services d'exécution au moment de l'exécution du processus. Les données relatives au workflow et aux applications doivent également être transférées entre les moteurs hétérogènes.

Chaque activité est associée à un domaine de workflow, cette caractéristique d'activité peut être mentionnée dans la définition du processus. Les décisions concernant le lieu d'exécution peuvent être plus complexes. On peut par exemple imaginer qu'une activité puisse être exécutée par un service particulier parmi d'autres et que cette décision soit faite lors de l'exécution. La sécurité, l'administration des systèmes, et le rétablissement après une erreur doivent être également considérés. Finalement, les services d'exécution de workflow devront partager les mêmes données qui normalement sont internes à chaque service ou même moteur de workflow.

La WfMC compte définir des niveaux de conformité, pour permettre à un produit qui répond à certaines spécifications de correspondre à certains scénarios d'utilisation. D'autres fonctions sont ajoutées ensuite pour répondre à des scénarios plus complexes.

Scénario 4 : L'exécution parallèle synchronisée Ce modèle permet à deux processus de s'exécuter indépendamment l'un de l'autre, par exemple dans deux services d'exécution différents, mais avec un point de synchronisation entre les deux processus. Cette synchronisation requiert que lorsque le point de synchronisation pré-défini est atteint par les deux processus, un événement commun est généré.

Dans le diagramme de la figure 3.17, la synchronisation se fait entre l'activité A3 du processus A et l'activité B4 du processus B.

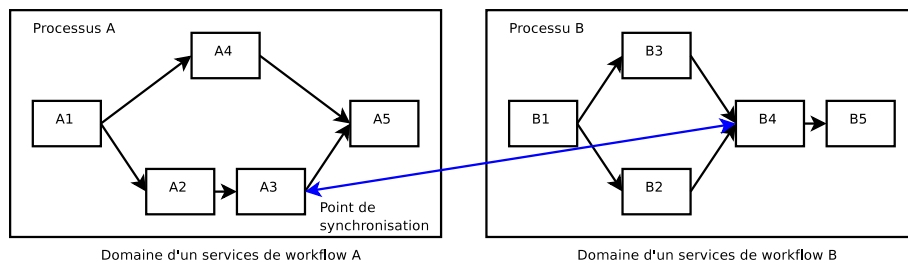


FIG. 3.17 – L'exécution parallèle synchronisée.

Les spécifications de la WfMC ne permettent pas encore de réaliser la synchronisation de telles activités.

3.4.5 Les API d'interopérabilité de services d'exécution de workflow

La nature des flux d'information entre systèmes de workflow hétérogènes est exprimée dans le diagramme de la figure 3.18 :

L'interopérabilité entre systèmes de workflow nécessite :

- une interprétation commune d'une définition de processus
- l'échange d'informations de contrôle lors de l'exécution et l'échange de données relatives aux workflows et aux applications entre les différents services d'exécution de workflow.

3.4.5.1 Une définition de processus à travers plusieurs domaines de workflow

Quand deux services d'exécution de workflow peuvent interpréter une même définition de processus, ils partagent une même vue des différents objets de la définition de processus (activités, applications, organisation et rôles, conditions de transitions, etc) et leurs attributs. Cela permet potentiellement de transférer l'exécution de certaines activités vers un autre service de workflow (scénario d'interopérabilité 3).

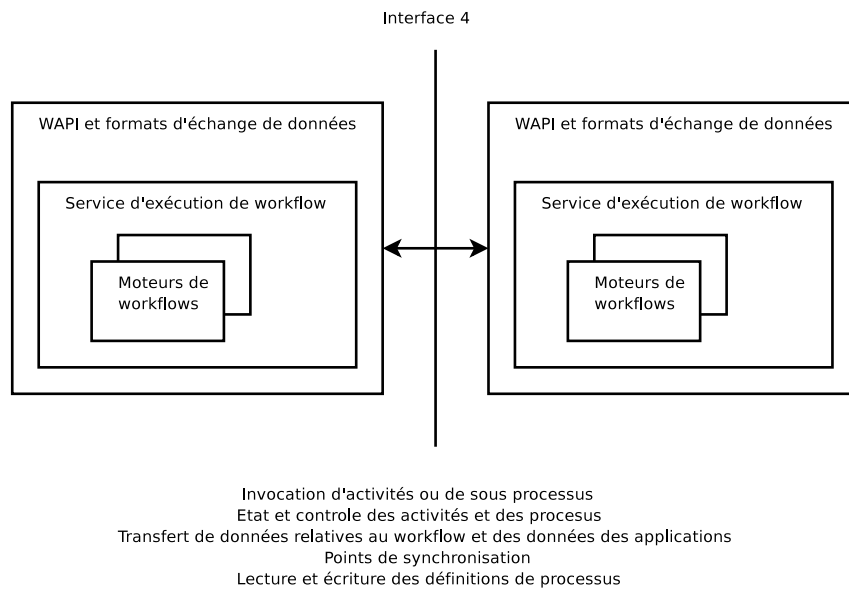


FIG. 3.18 – Les API d'interopérabilité de services d'exécution de workflow.

Si l'interprétation commune d'une définition de processus n'est pas possible, une alternative consiste en l'exportation des objets et attributs de la définition de processus pendant l'exécution. Cet échange est rendu possible par les API d'échange de définition de processus. Un moteur de workflow peut donc obtenir des données relatives à la définition de processus, et qui concernent une certaine activité (qui peut être un sous-processus) affectée à un certain service d'exécution de workflow.

Quand aucune de ces approches n'est possible, l'interopérabilité est soumise à une approche gateway dans laquelle une correspondance est faite entre le nom des objets et des attributs de chacun des environnements par l'intermédiaire d'une application gateway. Les deux environnements hétérogènes interprètent le même processus organisationnel, mais dans deux formats de définition de processus différents. On ne peut appliquer que des scénarios simple à ce niveau (scénarios 1 et 2).

3.4.5.2 Les interactions pendant l'exécution

A l'exécution, les appels WAPI sont utilisés pour transmettre le contrôle entre les différents services d'exécution de workflow pour déclencher l'exécution des activités sur un service spécifique.

Si les services supportent un niveau commun d'appels WAPI ainsi qu'une vue commune des objets de la définition de processus (comprenant les même conventions de nom et les données relatives au workflow et aux applications), les échanges d'information se font directement entre les moteurs de workflow. Toutefois, il faut un protocole sous-jacent commun pour le support des appels WAPI. Si ce n'est pas le cas, les appels WAPI peuvent être utilisés pour construire un gateway pour permettre aux services de workflow de travailler ensemble en faisant correspondre les différents objets et les

données entre les deux environnements et supporter les différents protocoles utilisés par les services d'exécution de workflow.

La figure 3.19 illustre les principales tâches d'une application gateway. Une activité d'un domaine A peut être mise en correspondance avec une activité du domaine B.

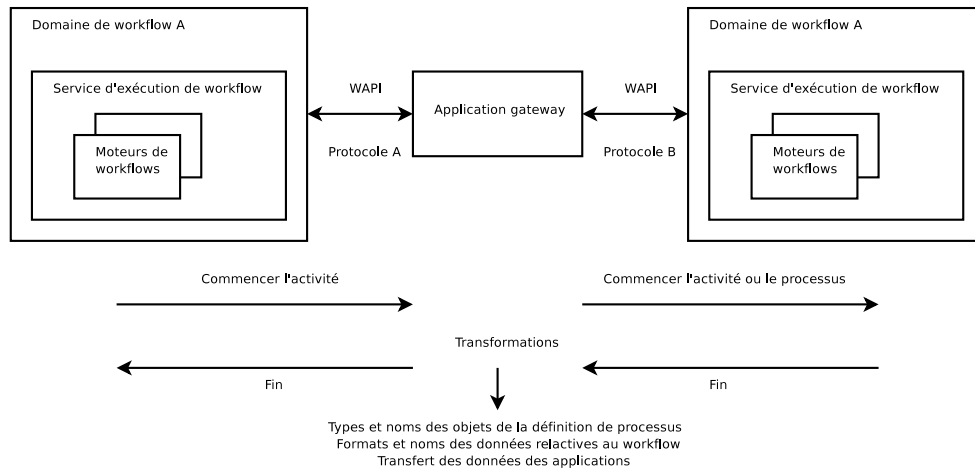


FIG. 3.19 – Une application gateway.

Beaucoup de commandes WAPI sont utilisées pour l'interopérabilité de systèmes de workflow, qu'elle soit faite directement entre les deux environnements ou indirectement par le biais d'une application gateway :

- l'établissement de session
- les opérations sur les objets et attributs d'une définition de processus
- les commandes sur le contrôle et l'état d'un processus
- la gestion des activités
- les opérations relatives à la gestion des données.

De plus, il faut qu'il y ait une administration commune au niveau de l'ensemble des domaines, c'est ce que développe l'interface 5.

Lorsqu'une activité est déclenchée sur un autre service d'exécution de workflow (le service subordonné), l'application cliente interagit toujours avec le service d'exécution principal (pour, par exemple, des requêtes sur le statut d'une instance de processus ou d'activité) et ces demandes doivent être relayées vers le service d'exécution subordonné à travers un ou plusieurs moteurs de workflow. Un service de notification d'événements est également nécessaire pour informer le service principal des changements de statut ou de l'achèvement des activités dont le système subordonné est responsable. Des interactions causées par des erreurs d'exécution et concernant le recouvrement du systèmes doivent aussi être envisagées.

La WfMC prévoit de développer d'autres WAPI pour supporter ces fonctions afin de permettre des scénarios complexes. Elle définira également des niveaux de conformité pour qu'une approche pratique d'interopérabilité puisse être envisagée.

3.4.6 Interface 5 : L'administration système

3.4.6.1 Les outils d'administration et de monitoring

Ce dernier domaine de spécification est une interface standard pour des fonctions d'administration et de monitoring permettant à une application de gestion d'un système de workflow de travailler avec d'autres moteurs de workflow. Les commandes concernant l'état d'une processus ont déjà été mentionnées dans les précédentes interfaces. Les fonctions d'administration et de monitoring concernent également les processus mais dans leur ensemble. Elles permettent une vue complète de l'état du travail dans l'organisation. D'autres aspects concernent la sécurité, le contrôle de l'ensemble des instances de processus et d'activités et les permissions d'accès.

3.4.6.2 Les API d'administration et de monitoring

La figure 3.20 montre une application de gestion indépendante qui interagit avec différents domaines de workflow. Dans d'autres cas, elle pourrait faire partie intégrante d'un service d'exécution de workflow particulier.

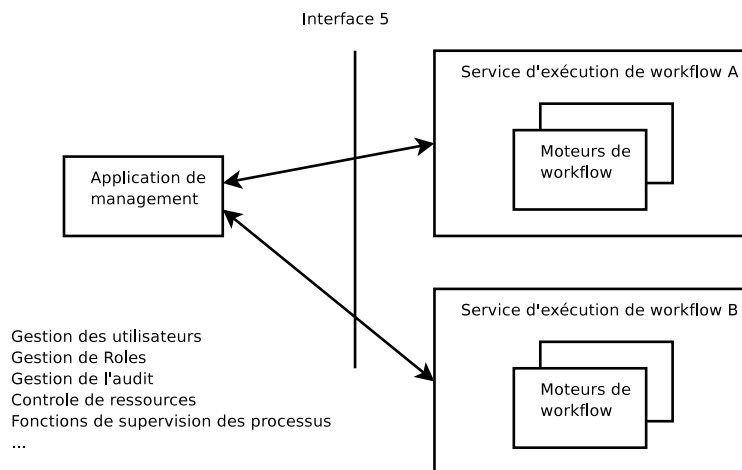


FIG. 3.20 – Une application d'administration.

D'autres fonctions de management peuvent être envisagées, comme la gestion des définitions de processus. L'application rangerait toutes les définitions dans un répertoire et le distribuerait dans les différents domaines de workflow en utilisant l'interface 1.

Pour le moment, les API d'administration comprennent des fonctions dans les domaines suivants :

- les opérations sur la gestion des utilisateurs : établissement, suppression, suspension, amendement de privilèges d'utilisateurs ou de groupes de travaux
- les opérations sur la gestion des rôles : définir, supprimer, amender une relation entre un participant et un rôle

- les opération de gestion d’audit : requête, imprimer, enregistrer, effacer des informations d’audit et d’événements
- les opérations de contrôle des ressources : modifier le niveau de concurrence des processus et activités, interroger les données de contrôle concernant les ressources
- fonctions de supervision de processus : changer l’état d’une définition de processus, rendre disponible ou non certaines versions de processus, changer l’état de toutes les instances de processus ou activité d’un même type, affecter des attributs à toutes les instances de processus ou d’activité d’un même type, interrompre toutes les instances de processus
- fonctions d’état de processus : ouvrir ou fermer une requête sur des instances de processus ou d’activité, récupérer des détails sur des instances de processus ou d’activité, groupées ou individuelles.

3.5 Conclusion

A ce stade, nous avons parcouru l’ensemble des termes liés au domaine du workflow, et ce à travers l’exploration d’un modèle de référence décrivant les composants et les fonctionnalités d’un workflow management system.

Le workflow reference model est un modèle mis au point voilà plusieurs années par la WfMC, mais ce modèle est toujours d’application de nos jours. Il décrit de manière très générale les composants d’un WFMS et les différentes fonctions qui leur sont associées, sans entrer dans des détails d’implémentation. Elle a une vision très large : de cette manière, la plupart des implémentations actuelles se reconnaissent dans ce modèle, et de nouveaux produits peuvent le considérer comme une base de travail.

La WfMC a également émis d’autres standards, notamment un langage de référence pour les définitions de processus, que nous verrons au chapitre suivant.

Chapitre 4

BPMN et XPDL

Le premier chapitre a présenté les notions de business process et workflow. Le second chapitre a défini les notions de base du domaine des workflows. Le chapitre 3 a introduit les WFMS (Workflow Management Systems) et le modèle de référence (émis par la Workflow Management Coalition) pour ces WFMS : les composants d'un WFMS et les cinq interfaces de la WAPI.

Ce chapitre présente deux manières de décrire un processus : de manière visuelle avec un diagramme de flux (le langage BPMN, standardisé par la BPMI) et de manière textuelle en utilisant le langage de description de processus XPDL (émis par la WfMC), décrit par un schéma XML. XPDL est portable, public, utilisable aussi bien par un moteur de workflow que par des outils de modélisation de processus. Les workflow patterns les plus courants peuvent être exprimés dans ces deux langages.

Après une brève introduction (section 1), les langages sont présentés : BPMN (section 2), XPDL (section 3), quelques workflow patterns dans ces deux langages (section 4).

4.1 Vers une définition visuelle d'un processus

La WfMC (Workflow Management Coalition) développe des spécifications pour les workflows depuis des années. Ces spécifications sont destinées aux développeurs de logiciels pour implémenter des solutions complètes et permettant l'interopérabilité avec d'autres systèmes.

Le groupe de travail 1 (importation et exportation de définition de processus) de la WfMC a donné naissance à XPDL (XML Process Definition Language), un langage permettant d'exprimer une définition de processus, c'est-à-dire la représentation d'un processus d'une organisation sous une forme permettant des manipulations automatisées. XPDL est donc utilisé lors de la modélisation ou lors de l'importation et l'exportation du processus par un service d'exécution de workflow. Cette définition de processus comprend un réseau d'activités et les relations qui les unissent, des critères de début et fin du processus, des informations sur chaque activité comme le participant, l'application qui lui est associée, ses données, etc., comme illustré par le méta-modèle de description de processus à la figure 4.1.

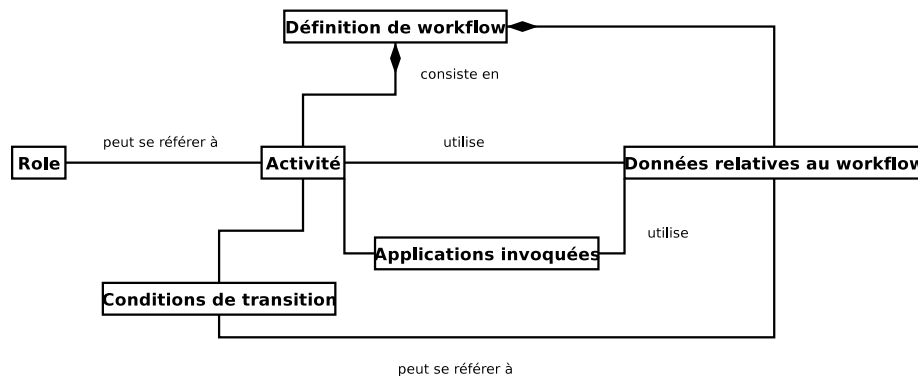


FIG. 4.1 – *Le méta-modèle d'une description de processus.*

Les processus peuvent être définis en XPDL grâce à un outil de modélisation, de manière qu'un moteur de workflow compatible avec XPDL puisse exécuter le processus. La définition de processus contient également des éléments relatifs à l'interface 5 (administration et monitoring).

La spécification XPDL définit les différentes activités d'un processus, la manière et l'ordre de leur exécution, mais ne définit pas la manière de visualiser cette définition de processus, d'une manière compréhensible pour un être humain. La WfMC n'a pas standardisé une notation graphique. Une telle notation graphique n'est pas destinée aux développeurs de logiciels de workflow, mais bien aux concepteurs de business process, dont le travail est justement d'exprimer un processus de leur organisation grâce à un diagramme de flux, qui sera "traduit" en XPDL par un outil de modélisation de processus.

Une définition de processus exprimée en XPDL peut donc être représentée (visualisée) différemment selon le logiciel de modélisation utilisé. Si plusieurs représentations (types de diagramme) sont utilisées, les personnes responsables du développement, de la maintenance, du management des processus de l'organisation auront donc une vision différente du workflow selon les multiples représentations. Il y a donc un "manque d'interopérabilité", en quelque sorte, pour ces personnes qui doivent opérer une traduction d'une notation graphique d'une définition de processus vers une autre notation (graphique ou non), selon l'environnement de workflow utilisé.

4.2 BPMN : Une notation standard pour les processus d'une organisation

La notation graphique d'une définition de processus n'étant pas destinée aux développeurs, elle n'est pas dans la portée de standardisation de la WfMC. Une autre organisation, la BPMI (Business Process Management Initiative), a dû standardiser une telle notation, en travaillant avec la WfMC.

La BPMI (Business Process Management Initiative) a développé la notation BPMN

(Business Process Modeling Notation). La version 1.0 a été publiée après 2 ans de travaux en mai 2004.

La BPMN est une notation qui a été conçue de manière à être compréhensible pour tout acteur de l'organisation, quel que soit son rôle : de l'analyste qui conçoit les diagrammes d'un processus de l'organisation au développeur responsable de l'implémentation et de l'exécution de ce processus, sans oublier les managers qui vont gérer et suivre le monitoring de l'exécution de ces processus. La notation BPMN peut être associée aussi à tout langage permettant d'exécuter un processus. Cette possibilité de traduire un diagramme décrivant un processus dans un langage utilisé par les WFMS forme donc le chaînon manquant entre le design de processus et son implémentation.

La notation BPMN définit donc une manière de dessiner un BPD (Business Process Diagram), elle est donc un dérivé de diagramme de flux (flowchart), qui est la manière la plus intuitive de créer un modèle graphique reprenant les opérations d'un processus d'une organisation. C'est donc un modèle qui caractérise un processus d'une organisation par des objets graphiques représentant d'une part des activités, et d'autre part les flux de contrôle définissant l'ordre d'exécution de ces activités.

Un BPD consiste en un ensemble d'objets graphiques familiers à ceux qui connaissent le flowchart diagram. Ces éléments se distinguent facilement entre eux par leur forme. Ces éléments fournissent à la fois un moyen simple de décrire un processus tout en reflétant sa complexité. Ces éléments graphiques ne sont pas très nombreux et appartiennent à une de ces quatre catégories :

1. les objets de flux
2. les objets de connexion
3. les couloirs (swimlanes)
4. les artefacts

4.2.1 Les objets de flux

Les objets de flux sont les objets principaux. Ils sont au nombre de trois.

1. Les événements (figure 4.2) : un événement est représenté par un cercle et représente quelque chose qui arrive pendant le processus. Il a une cause et un effet. Un icône au centre du cercle permettra de différencier les causes ou les résultats. Il y a trois sortes d'événements : start, intermediate, et end.

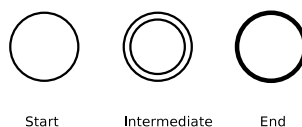
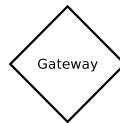


FIG. 4.2 – Les événements.

2. Les activités (figure 4.3) : une activité est représentée par un rectangle ovale et représente le travail effectué. Une activité peut-être de type "tâche" ou "sous-processus" (indiqué par un signe "+" dessiné en dessous du centre du rectangle). Dans ce dernier cas, un autre processus, défini ailleurs, est encapsulé dans cette activité.

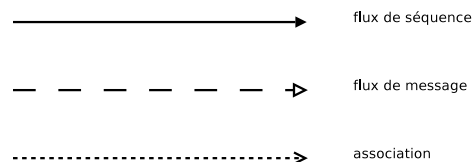
FIG. 4.3 – *L'activité.*

3. Les gateways (figure 4.4) : un gateway est représenté par une forme en diamant, et sert à marquer une décision, les convergences et divergence de flux (le comportement varie selon le marqueur).

FIG. 4.4 – *Le gateway.*

4.2.2 Les objets de connexion

Pour former un diagramme décrivant le processus de l'organisation, les objets de flux sont connectés entre eux grâce aux objets de connexion (figure 4.5).

FIG. 4.5 – *Les objets de connexion.*

1. le flux de séquence (ou flux de contrôle) : il sert à indiquer dans quel ordre auront lieu les activités dans un processus.

2. le flux de message : il montre un échange des messages entre deux participants. Ces participants seront représentés par deux pools.
3. l'associations : elle est utilisées pour associer des données, du texte et autre artefacts aux objets de flux. Elles sont utilisées pour montrer l'input et l'output des activités.

4.2.3 Les couloirs ("swimlanes")

Les méthodes de modélisation de processus utilisent un mécanisme appelé swimlanes (figure 4.6) pour organiser les activités en différentes catégories pour mieux faire ressortir certaines fonctions, rôles ou responsabilités. La BPMN reprend deux types d'objets pour construire ces swimlanes :

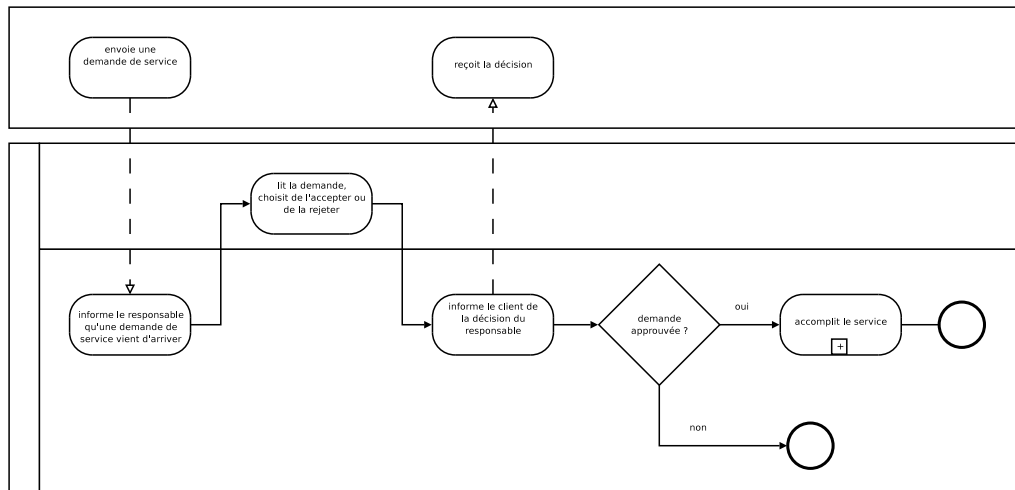
1. la pool : elle représente un participant dans un processus, et permet de partitionner les activités, les séparer en plusieurs pools dans un contexte B2B (business to business).
2. la lane : c'est une sous-partition (horizontale ou verticale) d'une pool pour organiser les activités.



FIG. 4.6 – Les couloirs (*swimlanes*).

Les pools sont utilisés pour séparer physiquement des participants ou des entités organisationnelles. Un flux de séquence ne peut dépasser les limites d'une pool d'activités. Les communications entre les participants (de deux pools différents) sont assurées par les flux de messages. Un processus est défini pour chaque pool. En dehors du fait qu'ils s'envoient des messages, ces processus sont indépendants. Chaque processus d'une pool a donc son activité de début et son activité de fin. La figure 4.7 reprend un exemple comprenant deux pools (donc deux processus) communiquant par messages.

Les lanes sont utilisées pour séparer les activités d'une même pool, selon le rôle ou la fonction de l'organisation associés à ses activités. Les flux de séquence peuvent traverser les limites d'une pool, mais un flux de message ne peut exister entre deux objets de flux appartenant à une même pool.

FIG. 4.7 – *Exemple de pools lanes.*

4.2.4 Les artefacts

BPMN permet aux concepteurs de processus une certaine flexibilité dans la notation, en vue de mettre en contexte certaines situations. Des artefacts peuvent être ajoutés à un diagramme, ils sont au nombre de trois (figure 4.8) :

1. Les objets de données : ce mécanisme sert à montrer quelle donnée est nécessaire ou est produite par une activité. Ces objets de données sont connectés aux activités grâce aux associations.
2. les groupes : ils servent à documenter sans affecter le flux de séquence.
3. Les annotations : elles servent à ajouter des commentaires textuels pour le lecteur du diagramme.

FIG. 4.8 – *Les artefacts.*

Les concepteurs de processus peuvent créer leurs propres types d'artefacts, par exemple en vue de caractériser l'input et l'output d'une activité. Les artefacts ne changent pas les activités, gateways, flux de séquence, etc... par leur présence.

4.3 XPDL : un langage XML de description de processus

Les éléments principaux du langage BPMN ont été exposés. On peut faire une correspondance entre des patterns graphiques rencontrés dans le langage BPMN et des patterns en code XML rencontré dans le langage XPDL. Avant de mettre en relation ces correspondances, donnons un aperçu de XPDL.

La WfMC (Workflow Management Coalition) a donné naissance à XPDL (XML Process Definition Language), un langage permettant d'exprimer une définition de processus, c'est-à-dire la représentation d'un processus d'une organisation sous une forme permettant des manipulations automatisées.

La WfMC a accepté la BPMN (Business Process Modeling Notation) comme notation pour XPDL et continuera à travailler avec le BPML pour prendre en compte les aspects présents dans les notations graphiques (ces aspects sont demandés par les concepteurs de processus d'organisation, qui utilisent la notation graphique pour décrire leur processus) mais absents dans les langages de description de processus comme XPDL, mais aussi d'autres comme BPEL4WS ou BPML. Il y a donc un "gap" entre la demande des utilisateurs de l'organisation, qui utilisent un certain niveau de détail dans leur notation graphique d'une définition de processus, et l'offre des producteurs de logiciels de workflow, dont la définition (en XPDL, BPML, ou BPEL, etc...) du processus n'atteint pas ce niveau de détail.

Voici quelques uns des éléments principaux utilisés dans un fichier de définition de processus exprimé en XPDL [16]. Beaucoup d'éléments ont été omis, non seulement pour avoir une vue d'ensemble de la définition, mais aussi parce qu'ils ne sont pas utilisés dans le cadre de ce mémoire. Nous travaillons donc avec une version "réduite" de XPDL. Les éléments sont présentés ici sous forme de liste structurée de manière à refléter la structure du fichier XML :

Package

- TypeDeclarations
- WorkflowProcesses
 - WorkflowProcess
 - Datafields
 - Participants
 - Applications
 - Application [FormalParameters]
 - Activities
 - Activity [Implementation ou Route, Performer, TransitionRestrictions]
 - Transitions

L'élément TypeDeclaration définit des types complexes (abstraits). Ces déclarations de types ont une portée qui s'étend à tous les workflows définis dans le fichier XPDL.

Les workflows sont définis par l'élément WorkflowProcess¹. Cet élément a plu-

1. Les éléments XPDL sont indiqués volontairement avec des majuscules, afin de mettre en évidence dans le texte le fait que ce sont des éléments du langage de description de processus XPDL.

sieurs descendants. Les DataFields reprennent les données relatives au workflow de ce processus. Les Participants reprennent les acteurs (ou plutôt le rôle qu'il auront à jouer) intervenant dans les activités du processus. Les Applications sont définies avec leurs Formal Parameters, qui définissent le type des paramètres d'entrée et de sortie de chaque application. Les activités sont définies par l'élément Activities. Les activités peuvent être des activités de travail (l'élément Implementation est présent) ou des activités de routage (l'élément Route est présent). Elles peuvent impliquer un acteur participant (l'élément Performer, qui référence un élément Participant défini plus haut) et des informations sur les transitions qui arrivent ou qui partent de cette activité (TransitionRestrictions).

Dans les sections suivantes, différents aspects seront détaillés : les types abstraits en XPDL, la déclaration des participants et la manière d'affecter un participant à une activité, la différence entre les paramètres formels (FormalParameters) et les paramètres actuels (ActualParameters), les activités normales (relatif à un travail) et de routage (une "fausse" activité).

4.3.1 Les types abstraits ou complexes

Dans le document XPDL du workflow, les types abstraits (complexes) sont définis le plus souvent dans un document séparé. Ce document est le plus souvent un document xsd, mais peut aussi être un document wsdl (si on utilise des web services comme application de support d'activités). Le schéma du type complexe peut également être incorporé directement dans le document XPDL.

Prenons en guise d'exemple le type "détail d'une commande". Il s'agit d'une liste (ou tableau, séquence, ...) d'éléments. Chaque élément est composé de deux parties : une partie pour le numéro du produit commandé (qui a donc comme valeur un nombre naturel) et une autre partie pour spécifier la quantité du produit commandé (encore un nombre naturel). Le type complexe est déclaré dans le document XPDL de cette manière :

```
<TypeDeclarations>
  <TypeDeclaration Id="detail_d_une_commande" Name="detail_d_une_commande">
    <ExternalReference
      location="http://www.exemple.org/xpdl/detail_d_une_commande.xsd" />
  </TypeDeclaration>
</TypeDeclarations>
```

Voici la représentation graphique d'un détail d'une commande (figure 4.9). La commande concerne n différents produits. La valeur de $npro\ i$ est le numéro d'un produit que le client veut acheter. La valeur de l'élément $qcom\ i$ représentent des quantités demandées de ce produit $npro\ i$.

Voici le schéma `detail_d_une_commande.xsd`. Il se trouve dans un document xsd annexe, mais pourrait être inclus dans le fichier XPDL dans la partie sur les déclarations de type, à la place de l'élément `ExternalReference`.

Voici un exemple d'instance de ce type :

```
<?xml version="1.0" encoding="UTF-8"?>
```

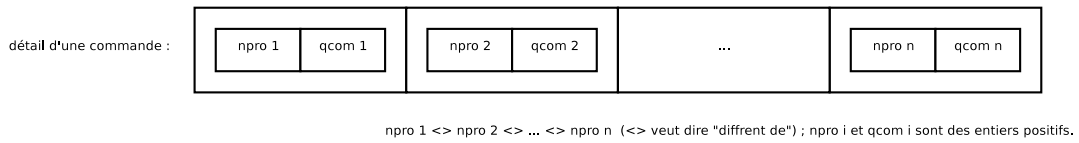


FIG. 4.9 – Le détail d'une commande.

```
<detail_d_une_commande ncli="61678" datecom="15/07/05">
  <element npro="1234" qcom="12" />
  <element npro="1243" qcom="13" />
  <element npro="1324" qcom="14" />
</detail_d_une_commande>
```

4.3.2 Les acteurs participants

Les acteurs doivent être déclarés dans la section définie par l'élément Participants. Les acteurs peuvent être définis de différentes manières. Une attention particulière est attribuée aux types human, role et system.

- RESOURCE_SET : un ensemble de ressources.
- RESOURCE : une ressource spécifique.
- ROLE : un acteur caractérisé par un rôle (une qualification) dans le contexte d'une organisation. Un rôle est pris en charge par un ensemble d'acteurs humains (HUMANS).
- ORGANISATIONAL_UNIT : un département d'une organisation.
- HUMAN : une acteur (exclusivement humain) qui interagit avec le système par le biais d'une application présentant une interface utilisateur.
- SYSTEM : un agent effectuant un traitement automatique.

```
<Participants>
  <Participant Id="Employe">
    <ParticipantType Type="ROLE">
      <Description>un employé de l'organisation</Description>
    </ParticipantType>
  </Participant>
</Participants>
```

La relation entre une activité et l'acteur qui va l'effectuer est faite grâce à l'élément Performer, fils de l'élément Activity. L'acteur en question doit être déclaré grâce à un élément Participant dans l'élément WorkflowProcess (la portée sera limitée à ce workflow).

```
<Participants>
  <Participant Id="Employe">
    <ParticipantType Type="ROLE">
```

```

    <Description>un employé de l'organisation</Description>
  </Participant>
</Participants>

```

A l'exécution, l'évaluation des acteurs potentiels peut avoir comme résultat un, plusieurs ou aucun acteur(s). Il appartient au fournisseur du WFMS de décider comment son programme réagira dans ces cas-là, car XPD ne fournit aucune information à ce sujet.

4.3.3 Les paramètres des applications

Dans les définitions des applications figurent les paramètres d'entrée et de sortie de ces application.

```

<Application Id="application_d_enregistrement_de_la_commande">
  <FormalParameters>
    <FormalParameter Id="detail_de_la_commande" Index="1" Mode="IN">
      <DataType>
        <DeclaredType Id="detail_d_une_commande" />
      </DataType>
    </FormalParameter>
    <FormalParameter Id="numero_de_reference_de_la_commande"
      Index="2" Mode="OUT">
      <DataType>
        <BasicType Type="INTEGER" />
      </DataType>
    </FormalParameter>
  </FormalParameters>
</Application>

```

Les paramètres formels Les paramètres formels (formal parameters) interviennent dans la définition des applications (l'élément Application) et des workflows (l'élément WorkflowProcess). Ils définissent le type requis pour les variables (ou les valeurs) à introduire comme paramètre de l'application. Le type peut être un type de base (entier, chaîne de caractères,...) ou un type abstrait (complexe) défini dans la section des déclarations de type (l'élément TypeDeclarations).

En plus du type de la variable à introduire, les paramètres formels définissent aussi le mode de passage des paramètres des applications. XPD supporte trois modes de passage de paramètres :

- IN : Il s'agit du passage par valeur, ou "read only". A l'appel de l'application, une nouvelle variable locale est créée et initialisée avec la valeur du paramètre. L'application travaille sur cette copie, celle-ci peut donc être modifiée sans que la variable d'origine introduite en paramètre soit affectée. A la fin de l'application, la copie est détruite.
- OUT : Il s'agit du passage par résultat, ou "write". A l'appel de l'application, une nouvelle variable locale est créée, mais n'est pas initialisée. La valeur du

paramètre importe donc peu. L'application modifie la variable locale et copie la valeur de celle-ci dans la variable introduite en paramètre à la fin de l'application.

- INOUT : Il s'agit du passage à la fois par valeur et par résultat, ou "read/write".

Les paramètres actuels Les paramètres actuels (actual parameters) sont les variables (contenant des valeurs) qui seront passées à l'application au moment où l'activité aura lieu, c'est-à-dire au moment de l'exécution (voir exemple plus haut).

4.3.4 Les activités

On peut considérer qu'il y a plusieurs sortes d'activités :

1. les activités implémentées : elles correspondent à un travail effectué
 - (a) soit par une application (avec un acteur humain, rôle, système... ou sans participant)
 - (b) soit par un sous-processus
 Cette application ou ce sous-processus sont définis dans les sections appropriées du fichier.
2. les fausses activités : ce sont des activités destinées à clarifier le code XPDL de manière à ce que sa structure évoque plus facilement les correspondances avec le diagramme BPMN qui lui correspond.

Exemple d'activité implémentée par un sous-processus :

```
<Activity Id="11" Name="Fill Order Subprocess">
  <Implementation>
    <Subflow Id="2" Execution="SYNCHR">
      <ActualParameters>
        <ActualParameter>orderNumber</ActualParameter>
        <ActualParameter>orderInfo.orderType</ActualParameter>
        <ActualParameter>orderInfo.emailAdress</ActualParameter>
      </ActualParameters>
    </Subflow>
  </Implementation>
</Activity>
```

Des exemples de fausses activités (les activités de routage) seront présentés dans les sections suivantes.

4.4 La correspondance entre BPMN et XPDL

XPDL est abordé ici de manière à faire correspondre un diagramme BPMN à un code XPDL. Autrement dit, un même workflow peut être exprimé de différentes manières en XPDL. Certains fichiers XPDL seront lisibles pour un lecteur humain car un diagramme exprimé en BPMN lui correspond et des correspondances entre un élément du diagramme BPMN et un élément du code XPDL pourront être faites. D'autres fichiers ne pourront être lus que par un programme.

On peut organiser le code XPDL de manière à faire ressortir tous les "work-flow patterns" [12] [13] présents dans le diagramme, à savoir le début et la fin du workflow, les activités de travail, les swimlanes (pool et lanes), les points de décision (branchage conditionnel), les AND-split, AND-join, OR-split, OR-join, les exceptions lors de l'exécution d'une activité, etc.

A ce stade, BPMN est beaucoup plus mûr que XPDL, au sens où on peut exprimer beaucoup plus de choses avec le diagramme qu'avec le code XPDL. Ces deux langages sont donc utilisés de façon restreinte de manière à garder ce qu'ils peuvent tous deux exprimer, en vue d'établir une correspondance entre ces deux langages.

4.4.1 Quelques patterns communs

1. Les éléments les plus simples sont les activités de début (figure 4.10), de fin (figure 4.11) et les transitions (figure 4.12) :



FIG. 4.10 – *L'événement marquant début de workflow.*



FIG. 4.11 – *L'événement marquant la fin du workflow.*

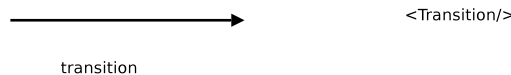


FIG. 4.12 – *Une transition.*

En fait, les "activités" de début et de fin sont en fait des événements traduits abusivement en activités de routage. Mais le but de cette section est de clarifier le code XPDL en établissant des correspondances entre des patterns graphiques et des patterns XPDL, de manière à ce qu'un lecteur humain puisse les lire.

2. Le deuxième pattern concerne les activités comprenant un travail effectué par un acteur et les activités implémentées par un autre processus (figures 4.13 et 4.14). Pour l'activité comprenant un travail effectué par un acteur et une application, l'application reliée à l'activité se spécifie grâce à l'élément Tool, et l'acteur (humain ou machine) responsable du travail est spécifié grâce à l'élément Performer.



FIG. 4.13 – Une activité comprenant un travail effectué par un acteur et une application.

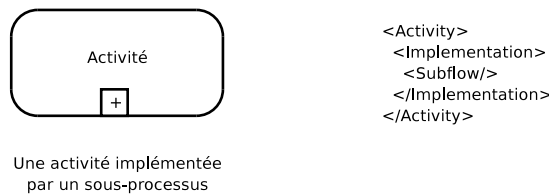


FIG. 4.14 – Une activité implémentée par un sous processus.

3. Les activités de routage, parmi elles le branchement conditionnel ("gateway" en BPMN, figure 4.15) et le AND-Split (figure 4.16).

Cette activité comprend un AND-Split et un élément implémentation, mais pour plus de clarté on peut diviser cette activité en deux : une activité d'implémentation et une activité de routage (`</Route>`), reliées l'une à l'autre par une transition.

4. De même, les AND-Join (figure 4.17) et OR-Join (figure 4.18) peuvent être définis en XPDL de la même manière.

On peut voir que le OR-Join est de type XOR pour XPDL. Le OR-Join est la suite logique d'un Gateway (branchement conditionnel), c'est-à-dire un Split de type XOR fait auparavant.

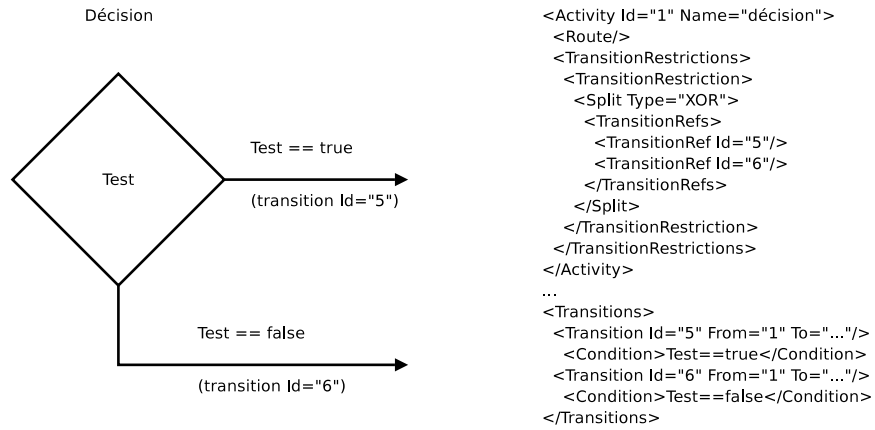


FIG. 4.15 – Un branchement "gateway".

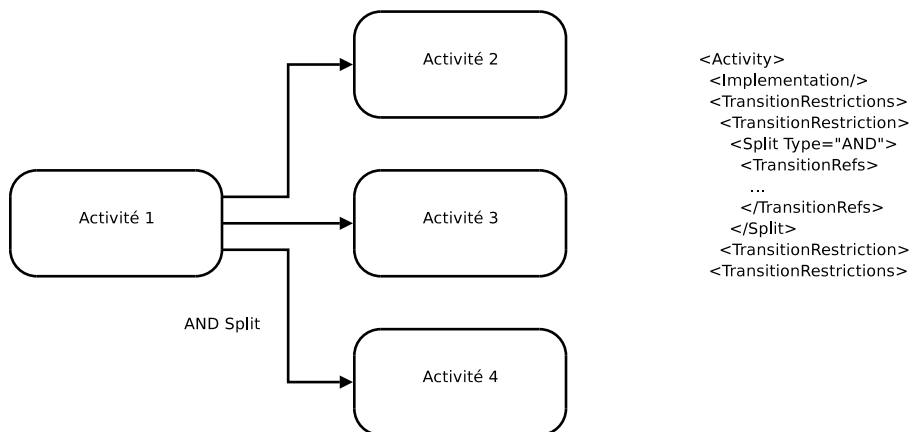
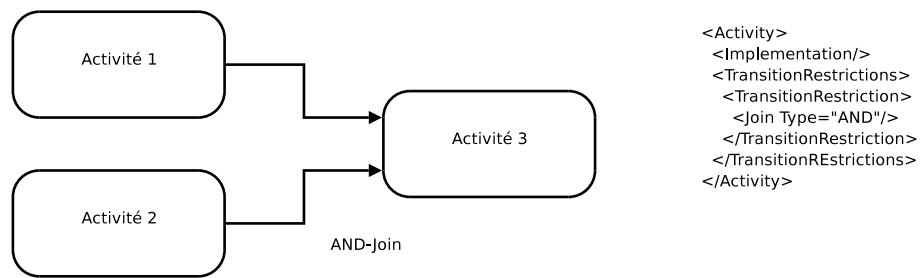
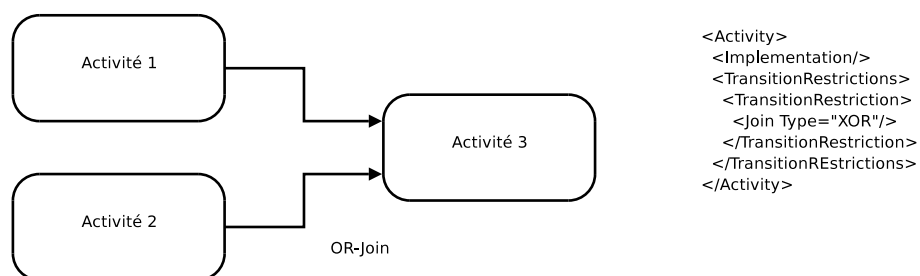


FIG. 4.16 – Un And-Split.

FIG. 4.17 – *Un And-Join.*FIG. 4.18 – *Un Or-Join.*

4.5 Conclusion

La notation BPMN (Business Process Modeling Notation) rencontre les demandes des concepteurs de processus. Les éléments de ce langage sont à même de décrire tous les aspects des processus du point de vue du modélisateur. Le rôle de ce langage est de dépeindre aussi précisément que possible un processus déjà implanté dans l'organisation ou d'en dépeindre un nouveau.

De son côté, le langage XPDL (XML Process Definition Language) est un langage de description de processus, plus précisément de son workflow. Il est utilisé aussi bien par les systèmes d'exécution de processus, qui importent et exportent ces définitions, que par les outils de modélisation de processus qui permettent au modélisateur de créer ces définitions formelles à partir d'un BPD (Business Process Diagram), c'est-à-dire un diagramme de flux ou d'activité, comme par exemple BPMN. Ce langage s'exprime en termes de workflow et de données assimilables par le système (des données XML sur les processus, les applications, les participants, les transitions et leurs conditions, les activités, etc.). Son rôle est centré sur l'automatisation du processus, et non pas sur sa conception et son analyse. Ce langage est donc moins riche au niveau de l'expression du processus que BPMN, mais est plus riche au niveau de la représentation des données du workflow.

Il y a donc un chaînon manquant entre les descriptions de processus faites par les modélisateurs de processus et les possibilités d'automatisation de ces processus offertes par les workflow management systems. La BPMI et la WfMC, les créateurs de BPMN et XPDL, travaillent ensemble pour que les possibilités d'automatisation offertes par le langage XPDL puissent correspondre à l'expression de processus complexes du monde réel.

Nous verrons au chapitre suivant un exemple de définition de processus dans un langage proche de XPDL, dont le code complet se trouve en annexe.

Chapitre 5

La construction d'un système de gestion de workflow

Le chapitre 1 a dressé le contexte dans lequel les organisations utilisent les business processes et les workflows. Le chapitre 2 a introduit les termes de base du domaine des workflows et les a défini précisément. Le chapitre 3 s'est consacré à l'architecture des systèmes d'exécution de workflow via la présentation d'un modèle de référence émis par la WfMC, le plus gros émetteur de standards du monde des workflows. Ce modèle était accompagné d'un ensemble de définitions. Le chapitre 4 a montré comment représenter une définition de processus à l'aide de BPMN (Business Process Modeling Notation), un BPD (Business Process Diagram) standard, ainsi qu'avec XPDL (XML Process Definition Language), un langage XML permettant de définir un processus sous une forme adaptée à des traitements automatisés de la part d'un moteur de workflow ou d'un outil de modélisation de processus.

Sur base de ces pré-requis, nous allons construire dans ce chapitre un nouvel outil de gestion de workflow. Il comprend un moteur d'exécution de workflow, une application front-end de présentation de liste de travaux à un participant, et une autre application front-end pour la gestion des définitions et des instances de processus (l'application de contrôle des processus). Ce moteur d'exécution de workflow va importer une définition de processus exprimée en XPDL. Cette définition de processus sera également construite dans ce chapitre à l'aide de BPMN, pour être ensuite "traduite" en XPDL.

La section 1 présente la définition du processus qui va servir de démonstration. C'est cette définition que le moteur de workflow développé dans ce chapitre va importer et exécuter. Ce mémoire n'a pas pour objet de définir une méthode de modélisation d'un processus, mais bien de développer un système d'exécution de workflow. Cette section présentera donc un processus simple, un cas d'école construit uniquement pour les besoins de ce travail, exprimé dans les langages BPMN et XPDL, qui ont été développés au chapitre 4.

La section 2 utilise le modèle de référence des systèmes de gestion de workflow (chapitre 3) pour présenter les différents composants du système à construire (le moteur d'exécution de workflow, l'application de gestion des listes de travaux et l'application de contrôle des processus), à l'aide d'un diagramme de composants et de déploiement

UML. La section 3 introduit les interactions entre l'utilisateur et les applications front-end (l'application de contrôle des processus et l'application de gestion de la liste de travaux) à l'aide de cas d'utilisation et de diagrammes d'activités de navigation, afin de bien cerner les différentes fonctions de ces applications front-end. Jusqu'ici, la description du système s'est faite à un haut niveau (architecture générale du système, description du processus à exécuter). Les dernières sections passent à la conception proprement dite de l'application : on parlera des technologies utilisées pour construire le moteur d'exécution de workflow et les applications front-end, des packages et des classes de ces applications dans les deux sections suivantes.

La section 4 effectue un travail préliminaire à la conception objet du système : une analyse du domaine, faite à partir du modèle de référence des systèmes de workflow. Il identifie et met en relation les objets du domaine (définition de processus, activité, instance de processus et d'activité, liste de travaux, ...), leur donne des attributs et les structure en packages. La section 5 développe, pour chaque composant identifié dans la section précédente, les classes et leurs méthodes, en s'inspirant de la WAPI (Workflow Application Programming Interface), et des objets métiers (analyse du domaine) récupérés dans le modèle de référence, à l'aide d'un diagramme de classes UML.

La section 6 décrit ensuite la technologie employée pour implémenter chacun de ces composants et leurs classes.

5.1 La définition du processus

Les méthodes de modélisation de processus pourraient faire l'objet d'un livre entier. L'objet de ce mémoire n'est pas de développer une méthode de modélisation de processus mais bien la construction d'un système de gestion de workflow pour exécuter ces processus en se basant sur leur définition. Cette définition étant obtenue suite à une phase de modélisation (figure 5.1).

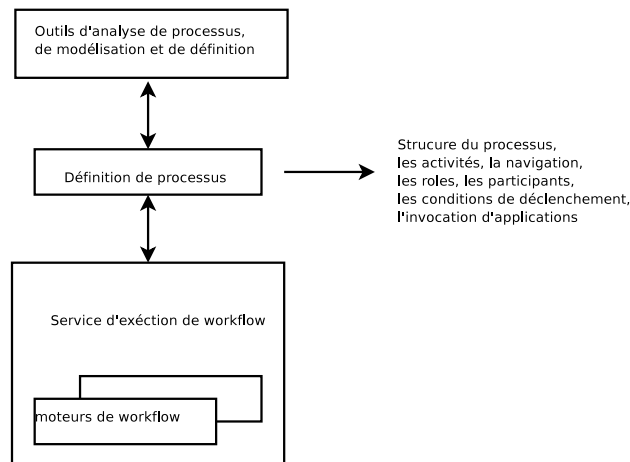


FIG. 5.1 – Interaction entre l'outil de modélisation et le système.

Pour ce travail, nous n'allons pas construire un outil de modélisation de processus ni en utiliser un tout fait. La définition de processus sera un cas d'école fait "à la main" pour les besoins de ce travail. Elle sera exprimée tout d'abord en BPMN puis traduite en XPD, puis elle sera importée et exécutée par le moteur d'exécution de workflow qui sera développé dans ce chapitre.

Les sections suivantes présenteront :

1. les participants,
2. les données relatives au workflow,
3. les activités,
4. les applications,
5. les données des applications.

Le code XPD du processus se trouve en annexe.

5.1.1 Les participants

La société "articles à vendre" vend des articles de la façon décrite par le processus "commande d'articles", qui comprend 3 participants :

1. un client,
2. un serveur,
3. un employé.

Le rôle du client est de commander des articles parmi une liste qu'on lui propose. La machine est un participant qui effectue des traitements automatisés (qui ne nécessitent pas de participation humaine) concernant l'enregistrement de la commande, la vérification du stock, des envois automatiques de message au client. Un employé est un rôle tenu par un des employés de la société, il est responsable du stock et de l'envoi des commandes.

5.1.2 Les données relatives au workflow

Il y aura plusieurs type d'objets qui seront manipulé par les activités.

1. le détail de la commande,
2. le numéro de référence de la commande,
3. le rapport sur le stock,
4. les informations liées au réapprovisionnement.

Le détail de la commande (figure 5.2) est une liste d'éléments. Chacun de ces éléments est lui même un objet reprenant le numéro d'un produit et la quantité commandée. C'est un type complexe qui sera déclaré dans la définition du processus.

Le numéro de référence de la commande est un nombre entier servant à identifier de façon unique une commande enregistrée dans la base de données de la société.

Le rapport sur le stock (figure 5.3) comprend des informations sur un ou plusieurs produits dont le stock est insuffisant. C'est un type complexe qui sera déclaré dans la définition du processus.

Détail d'une commande	
Numéro de produit	Quantité commandée
1256	2
4567	1

FIG. 5.2 – *Le détail d'une commande.*

Rapport sur le stock	
Numéro de produit	Quantité minimale à réapprovisionner
5678	5
1234	6

FIG. 5.3 – *Le rapport sur le stock.*

Les informations sur le réapprovisionnement (figure 5.4) comprennent des informations concernant le stock suite à un réapprovisionnement. C'est un type complexe qui sera déclaré dans la définition du processus.

Informations sur le réapprovisionnement	
Numéro de produit	Quantité disponible après réapprovisionnement
5678	5
1234	6

FIG. 5.4 – *Les informations sur le réapprovisionnement.*

5.1.3 Les activités

Voici les différentes activités effectuées par les participants :

1. Le client commande des articles : il sélectionne parmi une liste des articles qui l'intéressent et envoie sa commande au serveur,
2. le serveur enregistre la commande,
3. le serveur vérifie le stock : il vérifie si la commande peut être satisfaite avec l'état actuel du stock,
4. le serveur informe le client de l'insuffisance du stock,
5. l'employé réapprovisionne le stock,
6. le serveur informe le client que le stock est réapprovisionné,
7. l'employé envoie la commande,
8. le serveur informe le client de l'envoi de la commande.

5.1.4 Les applications

Dans ce processus, une application correspond à chaque activité. Si l'activité est effectuée par un participant humain, l'application comprend une interface d'interaction avec le participant. Voici la liste des application ainsi que leur paramètres :

1. l'application de commande des articles :
 - entrée : rien
 - sortie : un objet de type "détail de la commande" (type déclaré)
2. l'application d'enregistrement de la commande :
 - entrée : un objet de type "détail de la commande" (type déclaré)
 - sortie : un entier (le numéro de référence de la commande)
3. l'application de vérification du stock :
 - entrée : un entier (le numéro de référence de la commande)
 - sortie : un objet de type "rapport sur le stock" (type déclaré)
4. l'application d'information au client sur le manque de stock :
 - entrée : un objet de type "rapport sur le stock" (type déclaré), un entier (le numéro de référence de la commande)
 - sortie : /
5. l'application de réapprovisionnement de stock :
 - entrée : un objet de type "rapport sur le stock" (type déclaré)
 - sortie : un objet de type "information sur le réapprovisionnement" (type déclaré)
6. l'application d'information au client sur le réapprovisionnement de stock :
 - entrée : un entier (le numéro de référence de la commande), un objet de type "information sur le réapprovisionnement" (type déclaré)
 - sortie : /
7. l'application d'envoi de la commande :
 - entrée : un entier (le numéro de référence de la commande)

- sortie : /
- 8. l'application d'information au client de l'envoi de la commande :
 - entrée : un entier (le numéro de référence de la commande)
 - sortie : /

Voici le diagramme BPMN (figure 5.5 et 5.6) reprenant les activités et les données auxquelles les applications de ces activités accèdent :

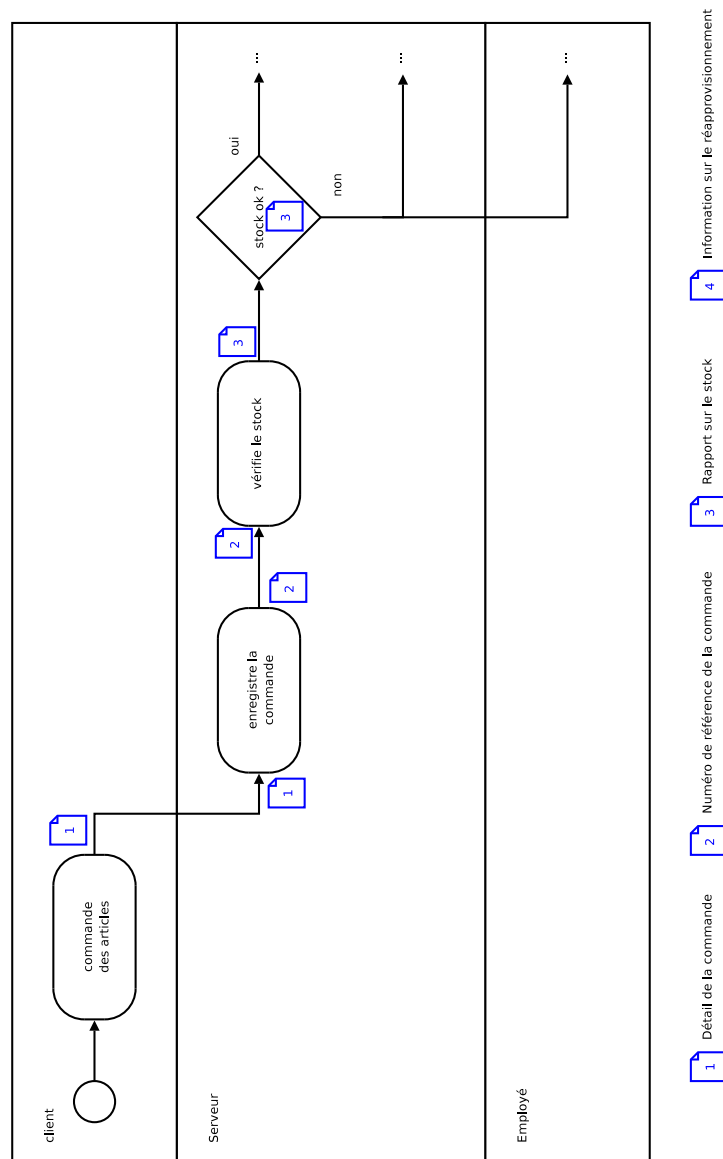


FIG. 5.5 – le diagramme BPMN et les données relatives au workflow (partie 1).

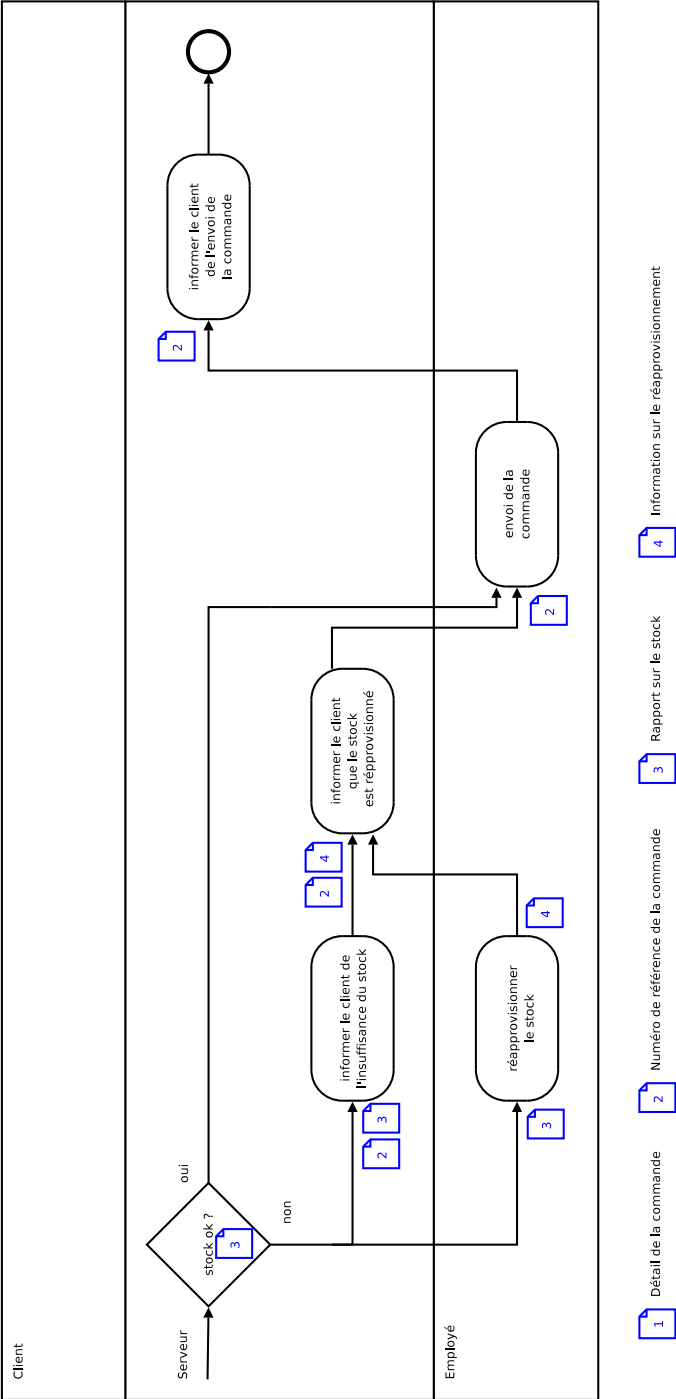


FIG. 5.6 – le diagramme BPMN et les donnees relatives au workflow (partie 2).

La traduction de ce diagramme BPMN en XPDL est donnée en annexe.

5.1.5 Les données des applications

En plus de définir le processus et les applications correspondant à chacune des activités, les applications auront des données internes qu'il faut également prendre en compte.

Les applications des activités seront reliées à une base de données contenant les données des applications :

- les produits proposés par la société "articles à vendre" et le stock de ces produits,
- les commandes en cours,
- le détail de ces commandes,
- les fiches client.

Voici un diagramme de classes UML (figure 5.8) représentant ces différents objets :

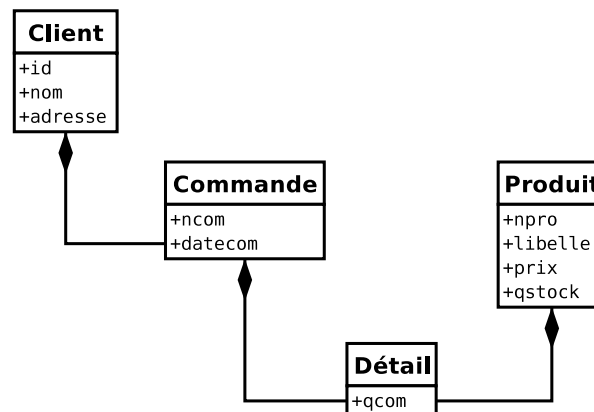


FIG. 5.7 – Les données des applications.

5.2 Les composants du système

La section 2, à partir du modèle de référence des systèmes de gestion de workflow (chapitre 3), présente les différents composants du système à construire (le moteur d'exécution de workflow, l'application de gestion des listes de travaux et l'application de contrôle des processus), à l'aide d'un diagramme de composants et de déploiement UML.

Au chapitre 3, le modèle de référence des systèmes de gestion de workflow décomposant le système en différents composants :

1. le service d'exécution de workflow,
2. les moteurs de workflow,

3. les application invoquées,
4. les applications clientes, qui comprennent des fonctions relatives
 - à la gestion de liste de travaux,
 - à la modélisation de processus,
 - au contrôle de processus,
 - aux fonctions d’administration et de monitoring.

Le système d’exécution de workflow présentait des interfaces (WAPI, Workflow Application Programming Interface) pour les fonctions suivantes :

1. l’importation et l’exportation de définition de processus,
2. la communication avec les applications clientes,
3. la communication avec les applications invoquées,
4. l’interaction avec d’autres services d’exécution de workflow,
5. la communication avec les outils d’administration et de monitoring.

Dans ce travail, les applications clientes ne comprendront pas la modélisation de processus et les fonctions d’administration, mais seulement la gestion de liste de travaux et le contrôle des processus. De même, ce travail ne s’inspirera que des fonctions relatives à la communication avec les applications clientes.

5.2.1 Les noeuds et les composants

Le système de gestion de workflow comprendra donc trois programmes principaux :

- le moteur d’exécution de workflow,
- le gestionnaire de liste de travaux,
- l’application de contrôle des processus.

Ce sont trois sous-systèmes qui s’exécutent sur des machines différentes, et sont reliées entre elles par un réseau, comme illustré à la figure 5.8 à l’aide d’un diagramme de composants UML.

En UML, un noeud représente une machine (ordinateur, serveur) sur lesquelles s’exécutent les composants du système de gestion de workflow, un composant représente la partie du système (le programme en exécution) qui s’exécute sur le noeud, il peut éventuellement être mis en relation avec des packages ou des classes, mais ce ne sera pas fait pour deux raisons : premièrement, le diagramme perdrait de sa lisibilité, deuxièmement, les classes et les packages n’ont pas encore été définis.

Il y a donc trois noeuds : le serveur d’application de gestion de liste de travaux, le serveur d’application de contrôle des processus et le serveur sur lequel tourne le moteur d’exécution des processus. Les applications front-end (les applications de gestion de liste de travaux et de contrôle de processus) comprennent des composants relatifs aux interfaces homme-machine (IHM) et aux traitements liés aux listes de travaux ou aux processus. Le moteur de workflow propose une interface aux applications front-end (l’interface service de gestion de workflow), comporte un composant de traitement des requêtes provenant des applications front-end et un composant qui implémente les fonctionnalités de gestion de données.

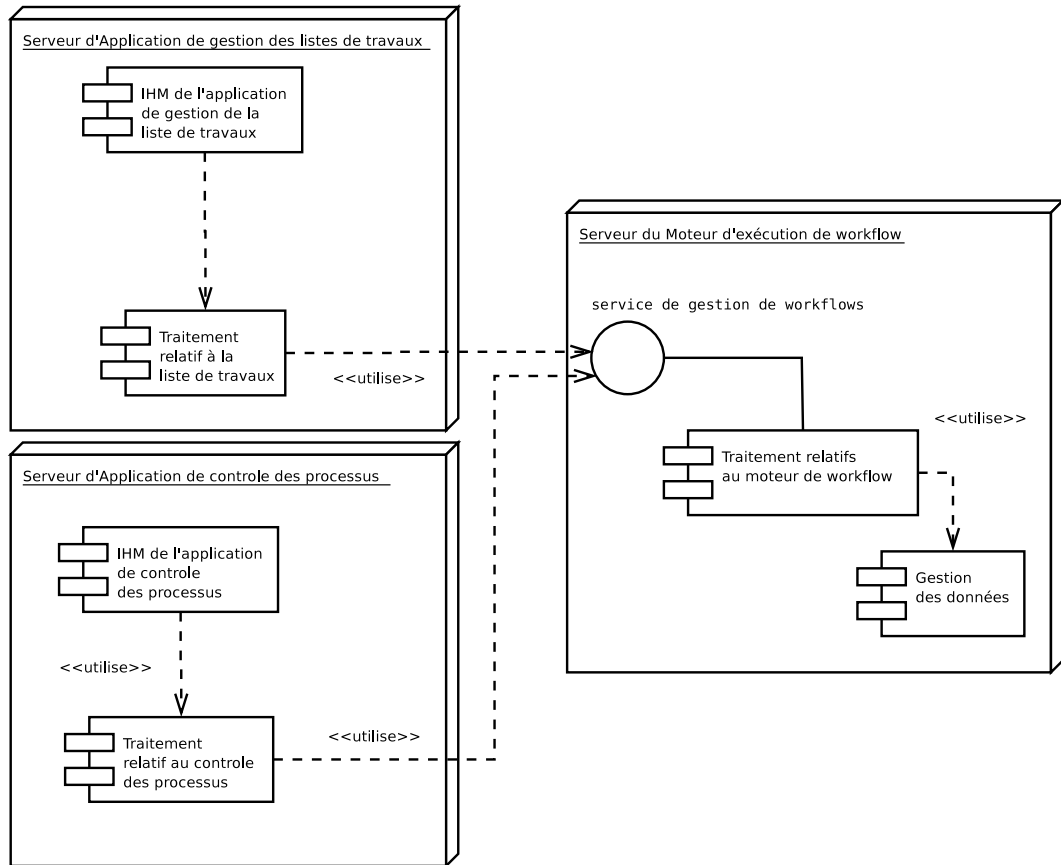


FIG. 5.8 – Les composants et les noeuds du système.

5.3 Les interactions entre l'utilisateur et les applications front-end

La section 3 introduit les interactions entre l'utilisateur et les applications front-end (l'application de contrôle des processus et l'application de gestion de la liste de travaux) à l'aide de cas d'utilisation UML et de diagrammes d'activités UML (pour décrire la navigation), afin de bien cerner les différentes fonctions de ces applications front-end.

5.3.1 Les cas d'utilisation

Les cas d'utilisation mettent l'accent sur les information qu'il faut montrer, à qui et pour quoi faire. Voici les étapes de la démarche :

1. L'identification des acteurs UML,

2. L'identification des cas d'utilisation (ainsi que les relations entre ces cas d'utilisation),
3. La structuration en package de cas d'utilisation.

5.3.2 L'identification des acteurs UML

Les acteurs UML pour le WFMS sont les suivants (figure 5.9):

1. Le Participant est une personne qui accède à l'application front-end de gestion de liste de travaux du WFMS pour recevoir ses éléments de travail et les effectuer.
2. Le Gestionnaire de processus: le gestionnaire de processus est une personne qui accède au WFMS (à l'application front-end de contrôle des processus) pour contrôler la liste des définitions de processus que peut exécuter le système, les instances de processus en exécution (il peut en créer de nouvelles), et contrôler l'avancement des travaux.

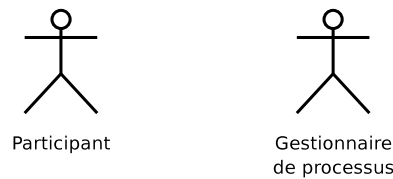


FIG. 5.9 – Les acteurs du système.

5.3.3 L'identification des cas d'utilisation

Pour chaque acteur UML identifié, les cas d'utilisation (figure 5.10) identifient les différentes intentions pour lesquelles il utilise le système. Ces intentions ont été préalablement exposées dans l'expression des besoins. Les acteurs et les cas d'utilisations seront ensuite représentés avec leurs associations (entre acteur et cas d'utilisation) et leur relations "extends" (entre les cas d'utilisation).

Pour le participant :

1. Consulter sa liste de travaux
2. Effectuer un élément de travail

Pour le gestionnaire de processus :

1. Consulter la liste de définitions de processus
2. Consulter la liste des instances d'un processus
3. Créer une instance de processus

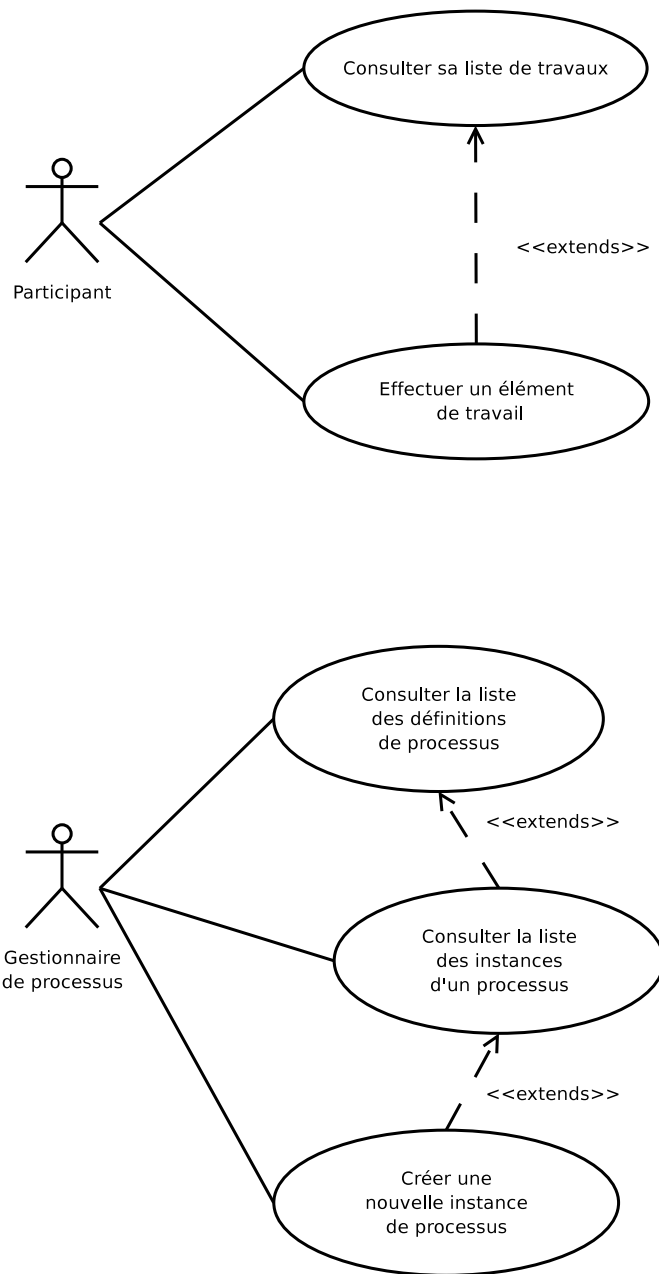


FIG. 5.10 – Les cas d'utilisation.

5.3.4 Structuration en packages de cas d'utilisation

Les cas d'utilisation sont ici regroupés en ensembles fonctionnels cohérents (voir figure 5.11). Les acteurs sont regroupés dans un package séparé sur lesquels s'appuient (par des relations de dépendances) les deux packages de cas d'utilisation.

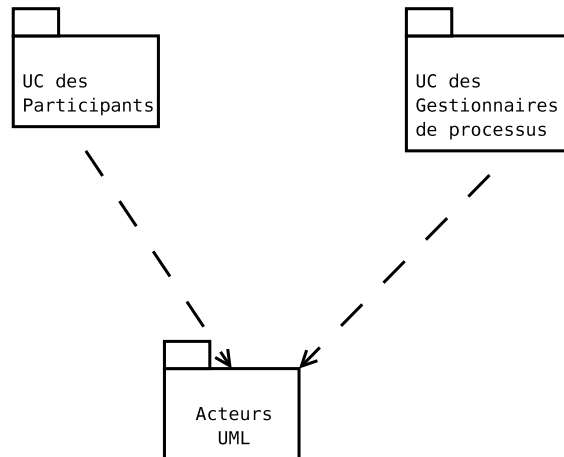


FIG. 5.11 – Les package de cas d'utilisation.

5.4 Modélisation de la navigation

Après avoir identifié les cas d'utilisation et avant de passer à la conception de classes d'analyse, il reste à étoffer les dialogues entre l'acteur de le système en identifiant les principaux écrans proposés aux utilisateurs et en réalisant des diagramme dynamiques représentant l'ensemble des chemins possibles entre ces principaux écrans.

Ces diagrammes s'appellent les diagrammes d'activité de navigation. Les diagrammes d'activité de base en UML rappellent fort ceux utilisés pour la notation BPMN dans le chapitre XPDL et BPMN. La notation de base pour ces diagrammes comprend :

- des activités,
- des transitions entre activités pouvant porter des conditions,
- des branchements conditionnels,
- un début et une ou plusieurs terminaisons possibles.

Cette notation de base est étendue pour modéliser plusieurs concepts différents, grâce aux conventions graphiques suivantes :

- une activité de "présentation" : elle présente un résultat à l'utilisateur,

- une activité de "action" : elle effectue un traitement sur des données en vue de préparer la réponse à la requête de l'utilisateur et de faciliter la présentation de cette réponse,
- une activité de "liaison" : elle renvoie à un autre diagramme d'activité, pour des raisons de structuration et de lisibilité.

Les noms des diagrammes utilisés lors des liaisons seront très proches de ceux des cas d'utilisation, mais les activités de liaison se rapportent bien aux diagrammes de navigation et non aux cas d'utilisation, bien qu'ils soient parfois structurés de la même manière.

Les diagrammes de navigations seront présentés suivant les cas d'utilisation et les acteurs UML. Nous retrouvons beaucoup d'éléments présents dans les précédentes étapes, mais ces diagrammes enrichissent les cas d'utilisations et leur spécification détaillée par leur intégration avec les écrans de la maquette d'IHM et le parcours des transitions entre les différents écrans. Ces transitions reprennent les dialogues détaillés dans les diagrammes de séquence avec des ajouts concernant la navigation entre les cas d'utilisation et les pré-conditions imposées à ces cas d'utilisation (par exemple, la connexion d'un acteur au WFMS).

Dans les diagrammes concernant le gestionnaire de processus, pour une question de lisibilité, l'action de déconnexion de cet acteur n'a pas été reliée par des transitions à chaque activité de présentation.

La modélisation de la navigation met en évidence deux types de classe pour les applications clientes :

- les classes de présentation, elles affichent à l'écran des informations à l'utilisateur,
- les classes d'action, elles traitent les requêtes reçues de l'utilisateur via l'interface et préparent les résultats de manière à ce que les classes de présentation aient un minimum de travail, uniquement lié à l'affichage.

Nous aurons donc deux acteurs UML : un Participant qui interagit avec l'application front-end de gestion des listes de travaux et un Gestionnaire de processus qui interagit avec l'application front-end de contrôle des processus. Pour chacun de ces acteurs (et donc chacune de ces applications), nous allons dresser un diagramme d'activité de navigation qui va caractériser les écrans et les fonctions que ces applications proposent aux acteurs.

5.4.0.1 L'application de gestion de liste de travaux

Les activités de présentation correspondent aux classes de dialogue de l'application de gestion de liste de travaux et sont les suivantes :

- l'affichage de la fiche de connexion et de la fiche "utilisateur non-trouvé",
- l'affichage de la liste de travaux.

Les activités d'action correspondent aux classes de contrôle de l'application de gestion de la liste de travaux et à leurs méthodes :

- l'action de connexion au système,

- l'action de déconnexion (normalement accessible depuis n'importe quelle activité de présentation, cela n'a pas été représenté pour une question de lisibilité),
- l'action de classement des travaux de la liste,
- l'action d'annonce du commencement d'un élément de travail,
- l'action d'annonce de la fin d'un élément de travail.

Parmi ces actions, il y en a une "invisible", c'est bien sûr l'obtention de la liste de travaux après la connexion du Participant au système.

Une activité de liaison relie un diagramme d'activité à un autre diagramme d'activité. Dans le diagramme 5.12, la liaison vers "effectuer un élément de travail" pointe vers un diagramme décrivant l'application invoquée servant de support à l'accomplissement de l'activité liée à l'élément de travail. Cette application invoquée de fait pas partie du WFMS mais de la définition du processus, ce diagramme n'est donc pas développé dans le cadre de la construction du WFMS.

5.4.0.2 L'application front-end de contrôle des processus

Les activités de présentation correspondent aux classes de dialogue et leurs méthodes de l'application de contrôle des processus et sont les suivantes :

- l'affichage de la fiche de connexion et de la fiche "aucun utilisateur trouvé",
- l'affichage de la liste des définitions de processus,
- l'affichage de la fiche d'une définition de processus,
- l'affichage de la liste des instances d'un processus,
- l'affichage de la fiche d'une instance de processus,
- l'affichage du formulaire de création d'une nouvelle instance de processus.

Les activités d'action correspondent aux classes de contrôle de l'application de contrôle des processus et à leurs méthodes :

- l'action de connexion au système,
- l'action de déconnexion (elle a été omise dans le diagramme pour plus de lisibilité),
- l'action de classement de la liste des définitions de processus,
- l'action d'affichage d'une définition de processus particulière,
- l'action de retour à la liste des définitions de processus,
- l'action d'affichage de la liste des instances d'un processus,
- l'action d'affichage de la fiche d'une instance particulière de processus,
- l'action de création d'une nouvelle instance d'un processus,
- l'action d'envoi des paramètres d'une nouvelle instance de processus.

Parmi ces actions, il y en a une "invisible", c'est bien sûr l'obtention de la liste des définitions de processus après la connexion du Gestionnaire de processus au système.

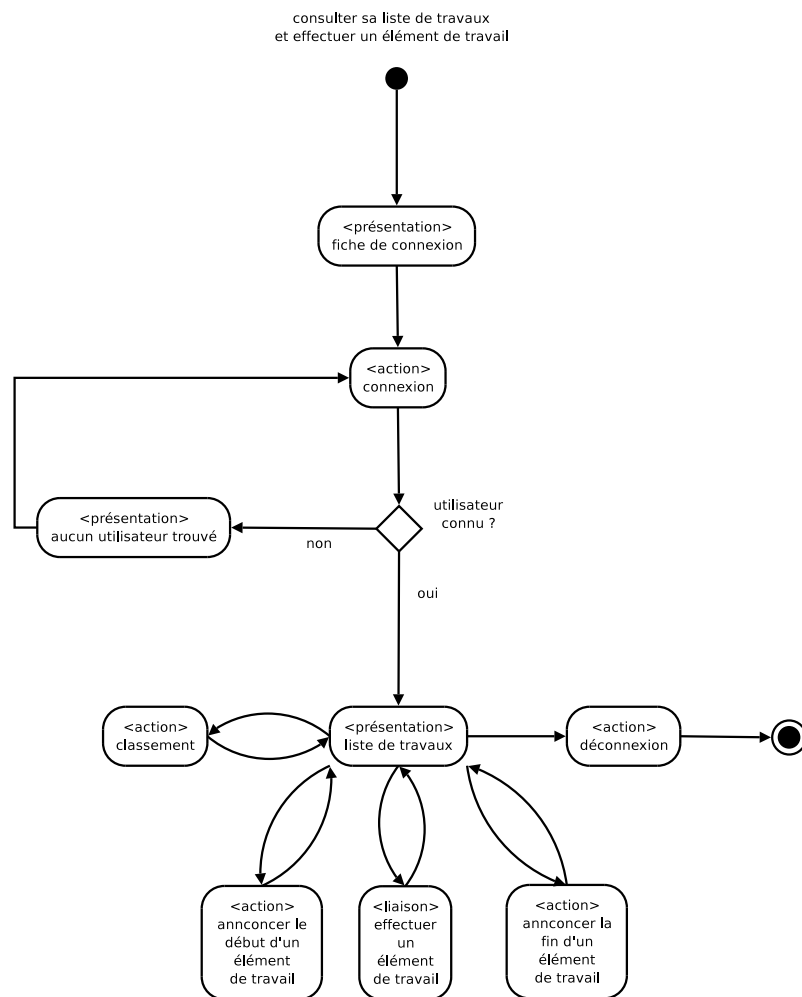


FIG. 5.12 – Navigation de la consultation de la liste de travaux.

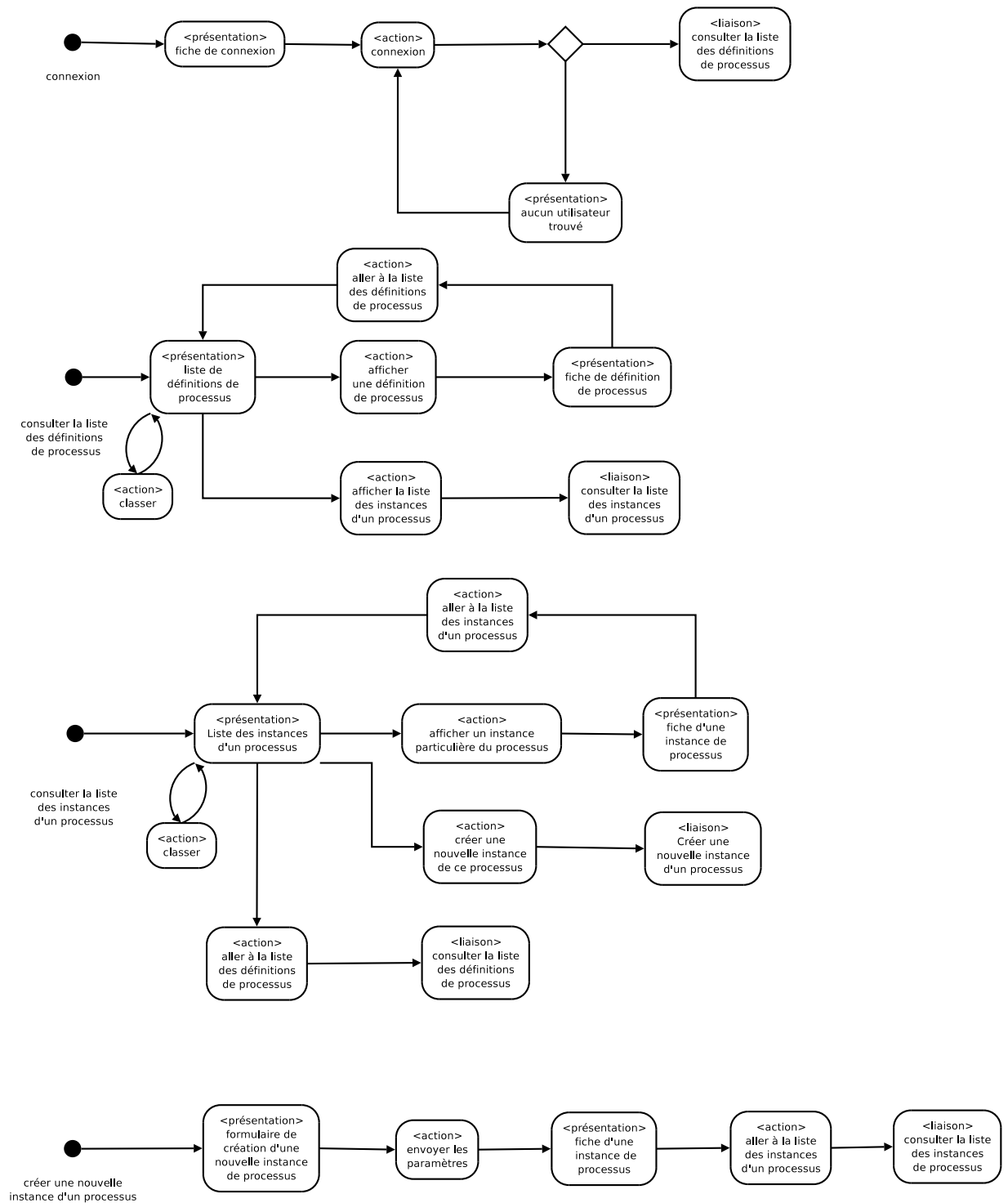


FIG. 5.13 – Navigation de la gestion des processus.

5.5 L'analyse du domaine

La conception objet du WFMS demande une description de la structure statique du système sous forme de diagramme de classes d'objets métier (c'est-à-dire les concepts manipulés par les experts du domaine des workflows). L'analyse du domaine conduit donc à un diagramme composé d'objets du domaine des workflows, des associations pour relier ces objets ainsi que des attributs pour caractériser ces objets. Ces concepts du domaine (les objets métier) peuvent être directement identifiés à partir des connaissances du domaine développées amplement par le modèle de référence des systèmes de gestion de workflow. L'expression initiale des besoins et les cas d'utilisation ne serviront que de sources complémentaires d'information.

Voici les différentes étapes de la démarche vers le modèle du domaine :

1. L'identification des concepts du domaine, de leurs attributs, de leurs associations
2. La structuration en packages de classes

5.5.1 L'identification des objets du domaine

Les objets du domaine sont les classes conceptuelles représentant des entités significatives du domaine, des représentations visuelles d'un objet du monde réel. Dans le cas du monde des workflows, cette analyse a déjà été effectuée par le modèle de référence des WFMS. D'autres informations peuvent aussi être récupérées d'autres chapitres, surtout en ce qui concerne les définitions de processus (dans le chapitre sur XPDL).

Voici les classes conceptuelles (figure 5.14) récupérées depuis la définition d'un processus. Pour de plus amples détails, les explications se trouvent dans le chapitre sur XPDL et BPMN.

Pour identifier le reste des objets métiers, les cas d'utilisation seront repris un à un et les concepts du domaine seront récupérés au fur et à mesure. Les cas d'utilisation seront passés en revue dans cet ordre : "Consulter sa liste de travaux", "Effectuer un élément de travail", "Consulter la liste de définitions de processus", "Consulter la liste des instance de processus" et "Créer une nouvelle instance de processus".

Le cas d'utilisation "Consulter la liste de travaux" concerne les objets métiers suivants :

- Acteur : l'Acteur est une personne susceptible de jouer un rôle (défini par l'élément Participant d'une définition de processus) pendant l'exécution d'une Instance de Processus. Un Acteur a un nom d'utilisateur et un mot de passe. Un Acteur peut avoir ou non le droit d'un gestionnaire de processus, c'est-à-dire de consulter la liste des Définitions de Processus et de créer une nouvelle Instance d'un Processus particulier.
- Participation : une Participation concrétise l'association entre un Acteur et une Instance de Processus dans lequel il intervient comme Participant. Dans le cas d'un Participant de type "humain" (l'attribut Type), un seul Acteur est nécessaire et suffisant pour la Participation. Dans le cas d'un Participant de type "rôle", il peut y avoir plusieurs Acteurs qualifiés pour ce rôle.

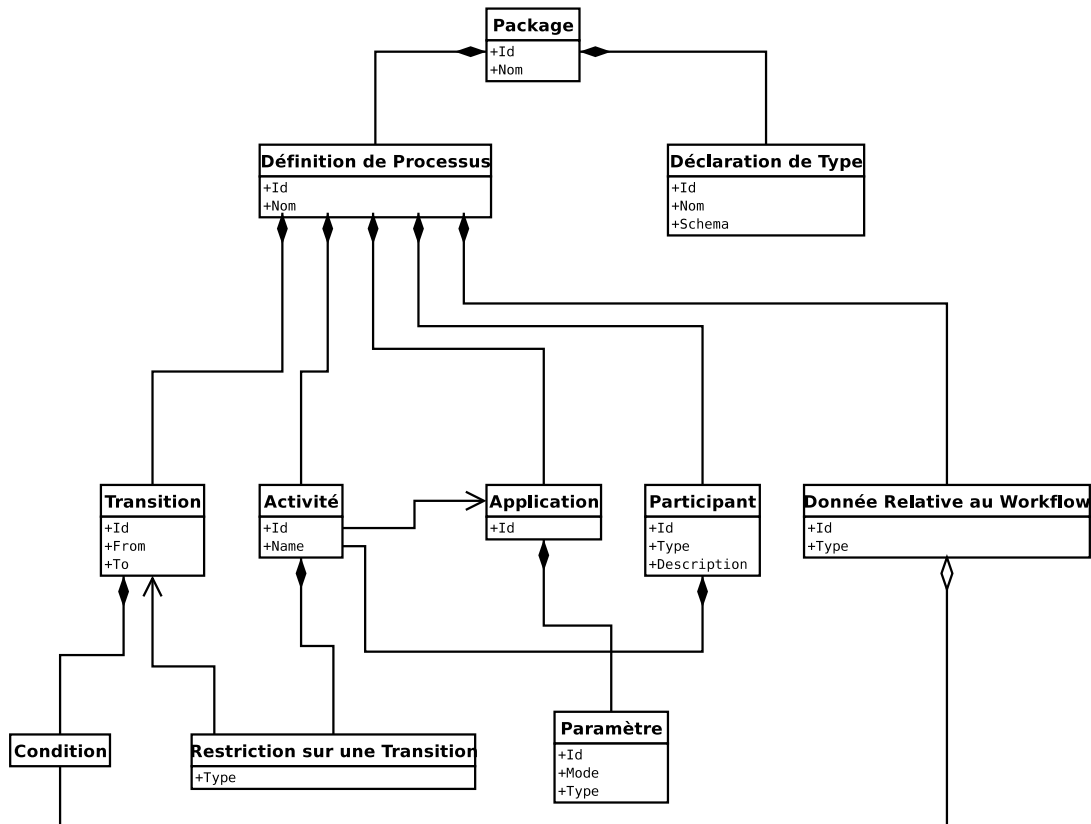


FIG. 5.14 – Objets liés à la définition d'un processus.

- Liste de Travaux : une Liste de Travaux est associée à un Acteur et contient ses éléments de travail.
- Élément de Travail.

Les cas d'utilisation "Effectuer un élément de travail" et "Consulter la liste des instances de processus" fait intervenir des concepts déjà rencontrés dans le modèle de référence des WFMS :

- déclaration des Applications (invoquées),
- déclaration des types de Données Relatives au Workflow,
- Instance de Processus,
- Instance d'Activité,
- Instance d'Application (invoquée),
- Instance de donnée relative au workflow.

Voici le diagramme (figure 5.15) comprenant ces derniers concepts et leurs relations :

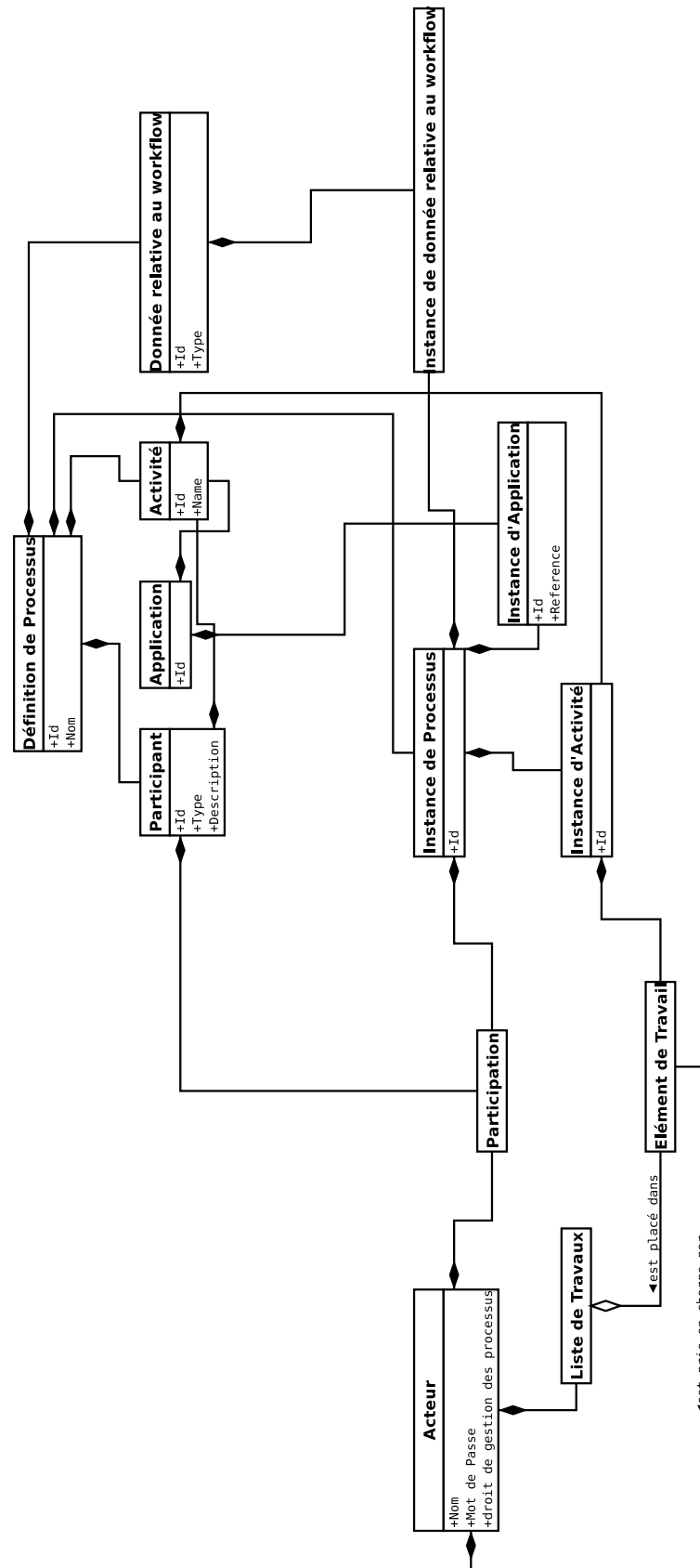


FIG. 5.15 – Les autres objets du domaine.

Les objets du domaine "Participant" et "Définition de Processus" étaient déjà présents dans le précédent diagramme. Six nouveaux concepts du domaines ont été ajoutés.

Un Acteur possède une seule Liste de Travaux. Un Élément de Travail est créé dans le cadre d'une Instance d'activité et peut être placé dans une Liste de Travaux, ou parfois plusieurs. Dans ce dernier cas, le Participant est de type "rôle" et l'élément de travail est placé dans la Liste de Travaux de tous les Acteurs enregistrés comme Participants, jusqu'à ce qu'un de ces Acteurs le prenne en charge.

5.5.2 La structuration en package de classes

Il y aura trois packages (figure 5.16) :

1. le package de définition des processus,
2. le package des instances de processus (et d'activités),
3. le package des listes de travaux.

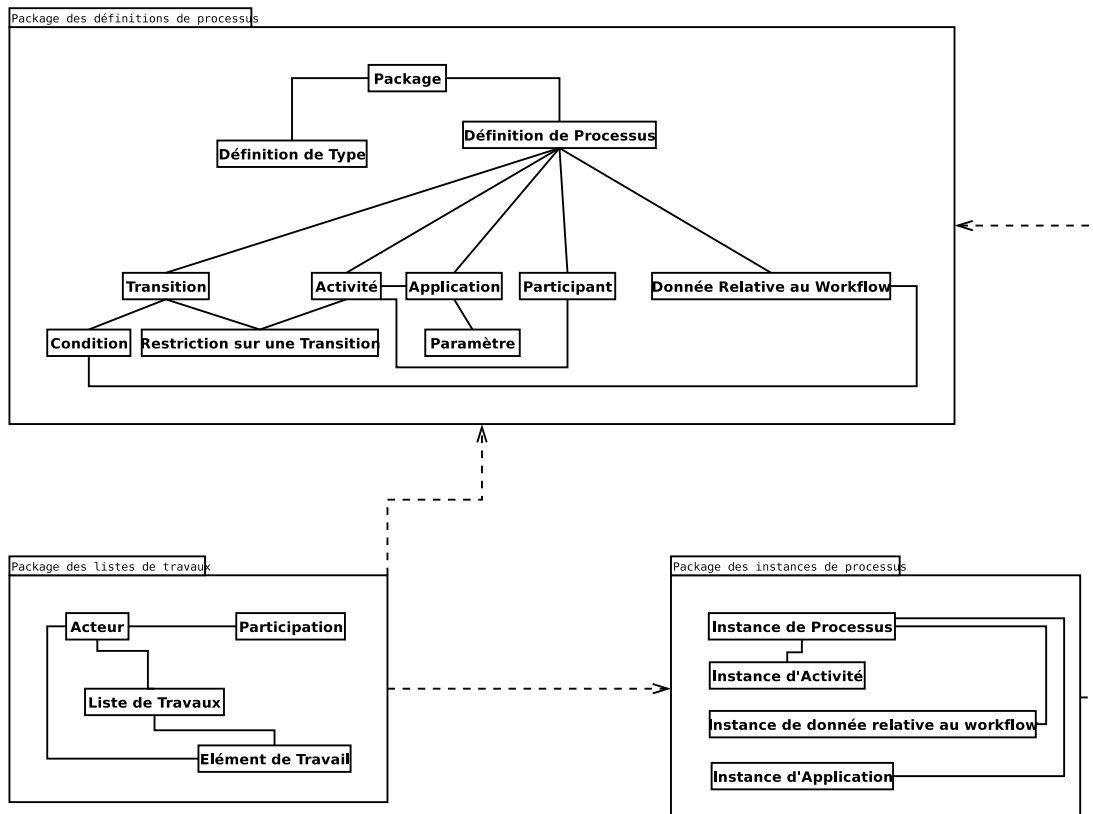


FIG. 5.16 – La structuration en package des objets du domaine.

5.6 Les classes de conception et d'implémentation

Cette section développe, pour chaque composant identifié, les classes et leurs méthodes, en s'inspirant de la WAPI (Workflow Application Programming Interface), des objets métiers récupérés dans le modèle de référence des systèmes de gestion de workflow, et des activités identifiées dans le modèle de la navigation à la section précédente. Les classes et leurs méthodes seront décrites à l'aide d'un diagramme de classes UML. La technologie employée pour implémenter chacun de ces composants et leurs classes fera également partie du sujet de cette section.

Ces classes peuvent être de trois types :

1. les classes de dialogues (ou frontière) : ce sont les classes d'interaction entre l'utilisateur et le système, typiquement les "écrans" proposés à l'utilisateur, ou les activités de "présentation" du modèle de la navigation des applications front-end.
2. les classes de contrôle : elles font la transition entre les dialogues et les classes métiers, permettent d'isoler les objets de l'interface et les données persistantes. Elles contiennent les traitements métiers et les comportements requis par les cas d'utilisation.
3. les classes d'entités : ce sont les objets métiers qui proviennent du modèle du domaine, ces données sont le plus souvent persistantes.

5.6.1 L'application de gestion de listes de travaux

D'après le modèle de la navigation de l'application de gestion de listes de travaux, il y a deux écrans pour l'application front-end de gestion de liste de travaux : l'écran de connexion (et de reconnexion, en cas d'erreur dans l'identification de l'acteur) et l'écran de la liste de travaux. Il y aura donc deux classes de dialogue correspondantes. Les fonctions relatives à la connexion, à l'obtention de la liste de travaux, à l'annonce du commencement et de la terminaison d'un élément de travail peuvent être retrouvées dans les classes de dialogue "connexion" et liste de "travaux" ainsi que dans la classe de contrôle "Liste de travaux".

Les classes de dialogue de ce diagramme appartiennent au composant "IHM de l'application de gestion de listes de travaux" et correspondent aux activités de présentation du modèle de la navigation de l'application de gestion de listes de travaux. Les classes de contrôle appartiennent au composant "traitement relatif aux listes de travaux" et correspondent aux activités d'action du modèle de la navigation de l'application de gestion de la liste de travaux. L'application de gestion des listes de travaux ne comporte pas de données persistantes (les listes de travaux se trouvent dans le sous-système du moteur d'exécution de workflow). Il n'y a donc pas de classes d'entité.

Les appels à ces fonctions seront relayés vers le moteur d'exécution de workflow, qui propose des fonctions inspirées de la WAPI, comme l'identification d'acteurs, l'obtention de la liste de travaux de cet acteur, et le commencement ou à la fin d'un élément de travail.

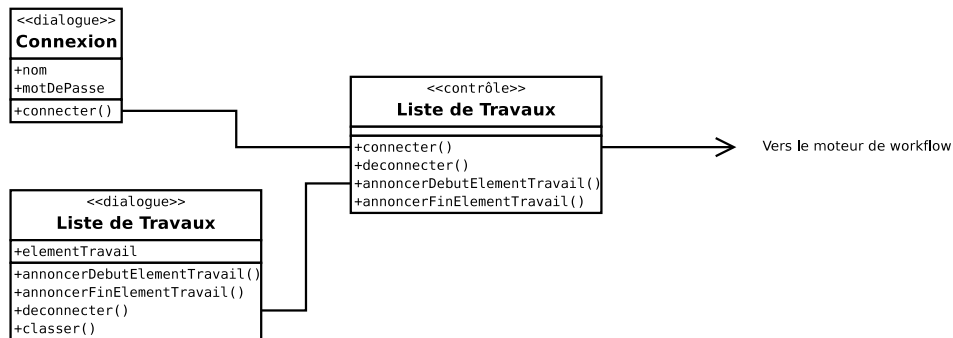


FIG. 5.17 – Les classes liées à l'application front-end de gestion de listes de travaux.

5.6.2 L'application de contrôle de processus

D'après la modélisation de la navigation de l'application de contrôle des processus, il y a six écrans pour cette application :

- la fiche de connexion,
- la présentation de la liste des définitions de processus,
- la fiche d'une définition de processus,
- la fiche de la liste des instances d'un processus,
- la fiche d'une instance de processus,
- le formulaire de création d'une nouvelle instance de processus.

A chaque écran correspond une classe de dialogue. Les activités d'action du diagramme d'activités de navigation sont reprises dans les classes de contrôle de l'application de contrôle des processus. L'application de contrôle des processus ne comporte pas de données persistantes (les données de contrôle se trouvent dans le sous-système du moteur d'exécution de workflow), il n'y a donc pas de classes d'entité.

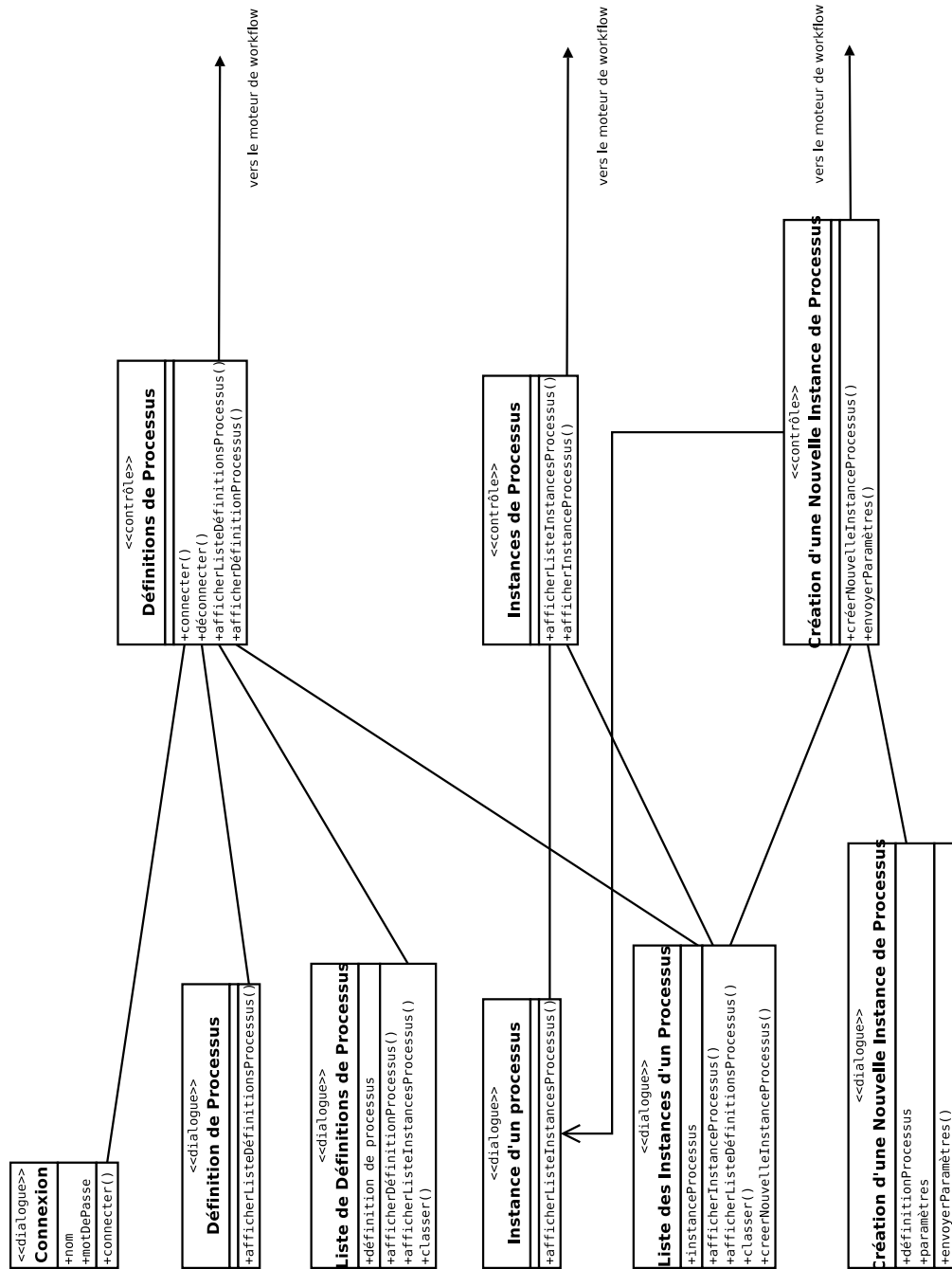


FIG. 5.18 – Les classes liées à l'application front-end de contrôle des processus.

5.6.3 Le moteur de workflow

Sachant qu'un "participant" (voir cas d'utilisation) est un utilisateur qui se connecte au système pour tenir le rôle d'un participant dans un processus, voici les fonctions que remplit le moteur de workflow pour l'application de gestion de listes de travaux :

- identifier le participant,
- obtenir la liste de travaux de cet acteur,
- recevoir de l'acteur le signal de commencement d'un élément de travail,
- recevoir de l'acteur le signal de fin d'un élément de travail.

Sachant qu'on appelle "gestionnaire de processus" un utilisateur ayant le droit de gérer les processus (définitions et instances), voici les fonctions que remplit le moteur de workflow pour l'application de contrôle des processus :

- identifier le gestionnaire des processus,
- obtenir la liste des définitions de processus pour l'acteur,
- obtenir les détails d'une définition de processus particulière pour l'acteur,
- obtenir la liste des instances de processus,
- obtenir les détails concernant une instance de processus particulière,
- créer une nouvelle instance de processus,
- assigner les paramètres pour cette nouvelle instance de processus.

On peut retrouver des fonctions similaires dans la WAPI (Workflow Application Programming Interface). Une classe de contrôle "Moteur de Workflow" (figure 5.19) implémente toute ces fonctions. Cette classe réside dans le composant "traitement relatifs au moteur de workflow" identifié dans une des sections précédentes.

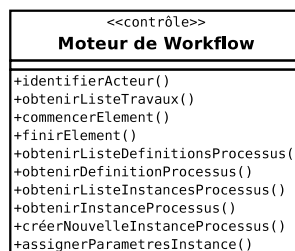


FIG. 5.19 – Les classes liées au moteur de workflow.

Dans l'analyse du domaine, trois packages de classes ont été identifiés :

- le package de définition des processus,
- le package des instances de processus (et d'activités),
- le package des listes de travaux.

Ces ensemble d'objets, leurs attributs, leurs associations et leur structuration en packages seront repris pour représenter les données persistantes du système de gestion

de workflow, les classes d'entités. Des classes de contrôle seront créées pour permettre de manipuler plus facilement les entités. Le moteur de workflow va accéder à ces données ou les modifier pendant l'exécution d'un processus grâce aux classes de contrôle suivantes :

- la classe de contrôle des définitions de processus,
- la classe de contrôle des instances (de processus, d'activité, ...),
- la classe de contrôle des listes de travaux.

Les classes de contrôle fournissent au moteur de workflow des méthodes qui serviront, lors de la création et l'exécution d'une instance de processus, à avoir accès et à modifier les données persistantes. Elles sont représentées à la figure suivante avec quelques-unes de leurs méthodes.

Par exemple, la classe de contrôle définitions de processus (la classe de contrôle DéfinitionProcessus) fournira les méthodes qui servent à :

- récupérer la liste des définitions de processus (enregistrée dans la base de données du moteur de workflow),
- accéder au détail d'un processus particulier (sa définition XPDL),
- récupérer, pour une définition de processus particulière, la liste des applications, la liste des participants,
- trouver la première activité du processus,
- trouver les activités qui succèdent à une activité donnée ou qui la précèdent,
- extraire la définition d'une activité particulière,
- extraire une condition de transition particulière,
- etc ...

La classe de contrôle des instances (la classe de contrôle Instance) fournit les méthodes qui servent par exemple à :

- récupérer des informations générales sur les instances d'un processus particulier,
- récupérer, pour une instance de processus particulière, des informations détaillées sur toutes les instances d'activité, instances d'applications ou instances de données relatives au workflow qui la concernent,
- ajouter la participation d'un certain acteur pour un certain rôle dans une instance de processus,
- enregistrer le fait qu'une définition d'application (dans la définition d'un processus) sera implémentée par une application dont on donne le chemin d'accès,
- créer des données relatives au workflow selon leur type spécifié dans la définition de processus et les rendre accessible aux applications de workflow,
- récupérer le détail d'une instance d'activité (son état, sa définition, son instance de processus) à partir de son identifiant.

Elle comprend aussi des méthodes plus générales qui fournissent des services comme :

- démarrer une instance de processus (changement d'état, création des instances d'activité, des éléments de travaux,...),

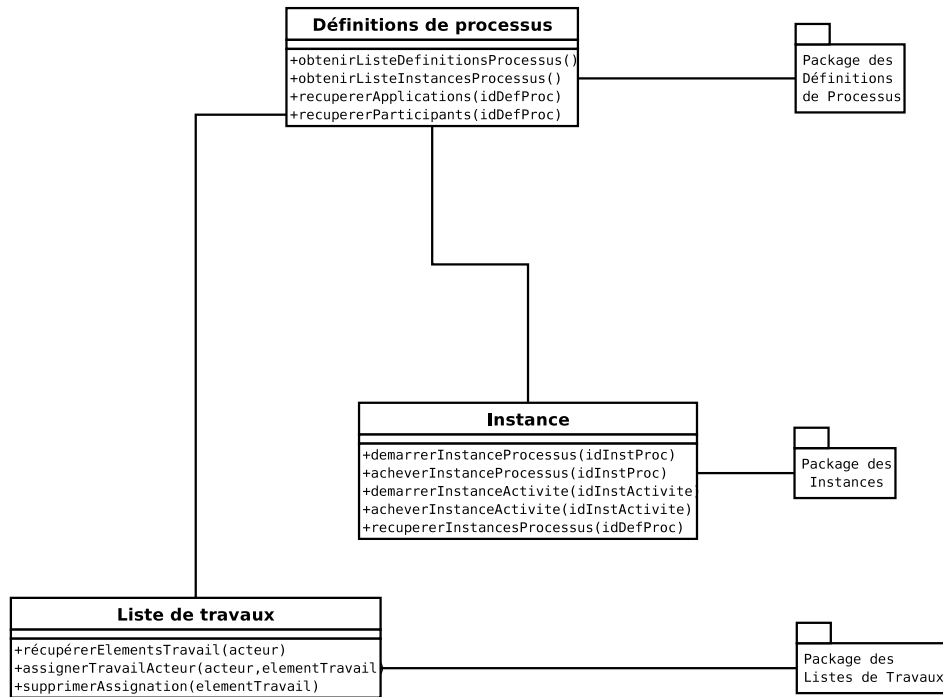


FIG. 5.20 – Les classes de contrôle du moteur de workflow.

- achever une instance de processus (changement d'état des processus et activités)
- démarrer une instance d'activité,
- achever une instance d'activité.

La classe de contrôle des listes de travaux (ListeTravaux) fournira des méthodes servant à :

- récupérer les éléments de travail dans la liste de travaux d'un acteur,
- enregistrer le fait qu'un acteur a commencé un élément de travail ou l'a terminé,
- etc ...

5.7 L'implémentation des composants et des données

Cette section présente la façon dont les composants sont implémentés. La figure 5.21 propose une vision globale de l'implémentation :

5.7.1 Le moteur d'exécution de workflow

Le moteur de système de gestion de workflow propose une interface pour le gestionnaire de liste de travaux et l'application de contrôle des processus. Ce moteur

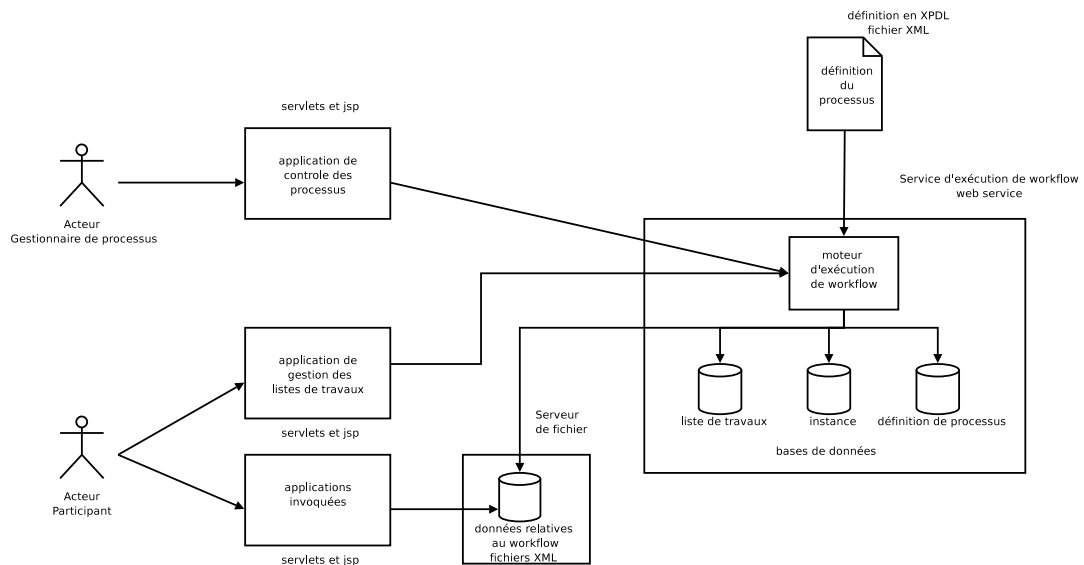


FIG. 5.21 – La vue globale de l'implémentation du système.

contient les données concernant la définition du processus, aux instances de processus et aux listes de travaux. Il a également accès aux données relatives au workflow sur un serveur distant. Nous avons donc un composant qui va publier des services pour des applications front-end. Le tout est d'exécuter le moteur de workflow sur un ordinateur d'un réseau, et de rendre ses services accessibles à d'autres programmes, s'exécutant sur d'autres ordinateurs.

Une des dernières évolutions dans le monde de l'informatique distribuée, adaptée pour notre cas, est le service web. Les services web sont globalement des fonctions, publiées sur un serveur. Ces services sont accessibles via un protocole standardisé : SOAP, un protocole basé sur XML. SOAP encode dans une enveloppe SOAP une représentation des objets ainsi que des appels et réponses de type RPC (Remote Procedure Call, appels procéduraux distants). Les règles d'encodage des enveloppes SOAP sont standardisées par les spécifications SOAP. En particulier, l'encodage des paramètres et la définition de leur type de données reposent sur des schémas XML. Les appels RPC, quand à eux, sont très proches de la syntaxe utilisée pour ceux d'un appel local de méthode. Les technologies de programmation distribuée comme Java RMI, COM de Microsoft et Corba utilisent toutes ce modèle.

Le moteur d'exécution de workflow est implémenté grâce à Glue (développé par webMethods), une implémentation de SOAP entièrement développée en java et peut être exécutée seule grâce à son serveur HTTP intégré. Elle est intéressante car elle propose des API très orientées programmeur, s'appuie sur le WSDL (Web Service Description Language), et est gratuite. En clair, elle permet à un objet java "moteur de workflow" et ses méthodes d'être publiés sur un réseau en tant que web service.

5.7.2 Les données relatives au workflow

Ces données sont définies dans un format public, accessible par le moteur de workflow et par les applications de workflow invoquées lors de l'exécution des activités. L'idée est donc de concevoir un format de données flexibles utilisables par un grand nombre d'applications. Ensuite, ces données devront être rendues accessibles aux applications et au moteur de workflow.

Pour décrire les données relatives au workflow, le langage XPDL utilise le même langage que SOAP : les schémas XML. Chaque type de donnée d'une description de processus est décrit par un schéma XML, utilisé par l'ensemble des applications voulant avoir accès ou modifier les données relatives au workflow.

De quelles logiciels dispose Java pour lire et modifier des documents XML ?

Sun propose JAXB (Java Architecture for XML Binding), un moyen efficace d'associer un schéma XML à un objet java (figure 5.22). JAXB permet aux développeurs d'incorporer plus facilement des traitements sur des données XML dans des applications. JAXB fournit, à partir du schéma des documents, des classes spécifiques reflétant les éléments décrits dans le schéma. Ensuite, à l'exécution, il transforme un document XML en un arbre d'objets construit avec ces classes Java. Cet arbre reflète les données contenu dans le document XML. Une fois l'arbre modifié, JAXB permet de sauver le nouvel arbre d'éléments sous forme de document XML.

Le moteur de workflow va utiliser JAXB non seulement pour lire les données relatives au workflow, mais également pour lire les informations contenues dans la description de processus écrite dans le langage XPDL.

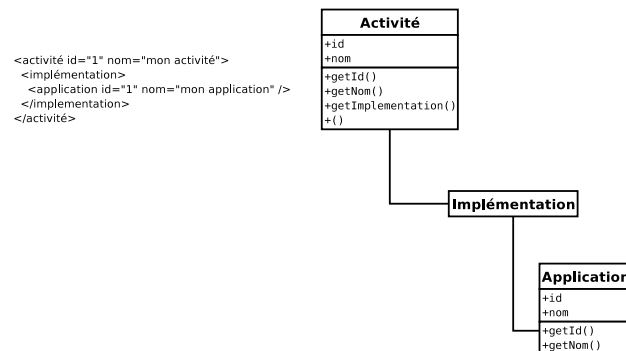


FIG. 5.22 – Utilisation de JAXB pour représenter des fichiers XML sous forme d'objets.

5.7.3 Les applications front-end

Les applications front-end (le gestionnaire de listes de travaux et l'application de contrôle des processus) vont récupérer le fichier WSDL publié par le web service "moteur de workflow" et invoquer les services distants grâce aux API proposés par

Glue. Ils devront également être capables de lire et modifier des document XML, décrits à partir d'un schéma. Les applications front-end seront, comme les autres composants, développées en java.

Ces applications utilisent les technologies servlets et les jsp. Ce sont deux produits fournis par Sun, implémentés en Java, qui vont permettre, depuis un navigateur web, d'interagir avec les autres composants : les web services du moteur de workflow, le serveur de fichier XML des données relatives au workflow.

5.8 Conclusion

Cette implémentation d'un workflow management system, ainsi que la description et la définition formelle d'un processus destiné à être exécuté par ce système, sont basées sur plusieurs standards de référence dans le monde des workflows :

1. le modèle de référence des systèmes de workflow, qui définit des termes communs et une architecture commune pour les WMFS,
2. le langage de description de processus XPDL,
3. les diagrammes de business processus BPMN.

Les fonctions proposées par ce système concernent la gestion des listes de travaux et le contrôle des processus et de leurs instances, mais le modèle de référence en prévoit également d'autres : des fonctions de manipulation des données relatives aux définitions de processus et des fonctions de monitoring et d'administration.

Les processus sont directement interprétés depuis leur définition XPDL, et les instances des processus et des activités sont maintenues par le moteur d'exécution de workflow dans une base de données.

Les données manipulées par les participants via certaines applications sont des données XML. Ce sont des schémas qui décrivent ces données, les applications utilisent ces schémas pour pouvoir lire modifier ces données.

Tout comme d'autres implémentations, comme par exemple workspaces [workspaces], elle se base sur des technologies web pour la communication entre les composants et sur XML pour la représentation des données, aussi bien les définitions des processus que les données relatives au workflow.

Les livres [8] et [5] ont servi de référence pour les diagrammes UML.

Chapitre 6

Conclusion

Ce mémoire retrace les origines des business processes et des workflows, en expliquant comment et pourquoi les organisations ont adopté ces concepts. Il définit ensuite les termes business process et workflow, ainsi que les termes associés à ces définitions. Il présente également le modèle de référence des workflows, qui est sans doute la publication la plus importante du domaine. Elle permet d'explorer en détail les concepts, les composants et les interfaces d'un système d'exécution de workflow standard. Ce modèle peut aussi être considéré comme base pour une analyse du domaine. Il se penche ensuite sur la définition d'un processus (visuelle d'une part, et formelle d'autre part) en présentant les langages BPMN et XPDL. Il propose enfin un travail pratique sur base du modèle de référence et des langages de définition de processus. Ce travail pratique comprend la construction d'une définition de processus originale (un cas d'école), exprimé dans les langages BPMN et XPDL, ainsi que le développement et l'implémentation d'un WFMS sur base du modèle de référence, qui exécutera le processus.

Les standards définis pour les workflows présentés dans ce mémoire sont toujours en cours d'élaboration. De plus, ce mémoire ne tient compte que du modèle de référence et du langage XPDL, mais il y a bien d'autres spécifications et d'autres projets dans le monde des workflows. Par exemple, le projet JSR 207 (Process Definition for Java), créé par le JCP (Java Community Process) en mars 2003, propose une syntaxe pour définir les business processes et utilise le Java Language Metadata (JSR175) pour relier les données et les activités de cette définition à du code Java (à des variables, classes). Il est un composant du logiciel Weblogic Workshop 8.1, qui permet d'assembler de manière visuelle des applications web, des web services, des JSP, des EJB ou des business processes en un grand processus global, en permettant au développeur de naviguer à travers un diagramme de processus et prendre connaissance du code source correspondant aux activités.

On peut en citer d'autres spécifications, sans entrer dans les détails :

- BPSS (Business Process Specification Schema, qui fait partie de la suite de spécifications ebXML [www.ebXML.org]) est un standard permettant la collaboration entre des systèmes de management de processus;
- BPEL, qui est une spécification basée sur l'échange de message, et peut être vu comme un langage XML décrivant un flux de contrôle formé d'une combinaison

de web services.

- Le diagramme d'activité UML, qui peut servir à représenter visuellement des Business Process,
- PIP (Partner Interface Processes) de RosettaNet [3], qui sert à construire des processus entre différents partenaires en définissant des dialogues, entre systèmes, basés sur XML.

Il y a également un grand nombre de produits de workflow disponibles, utilisant des standards différents. Une liste des produits basé sur le modèle de référence ou utilisant XPDL est disponible sur le site de la WfMC [4]. Le monde des workflow comprend donc un nombre impressionnant de spécifications et de produits.

Dans ces différents produits et spécifications, l'emploi des technologies web et de langages basés sur XML est justifié par le fait que les systèmes de workflow sont de plus en plus destinés à être interopérables, et doivent donc reposer sur des technologies standards et publiques. Ces produits et ces spécifications peuvent donner une idée des pistes non prises en compte dans ce mémoire.

Une amélioration pour le travail pratique de ce mémoire serait de permettre, comme présenté dans le modèle de référence, l'exécution du processus en utilisant plusieurs moteurs de workflow. L'exécution de ce processus serait partagée entre plusieurs de ces moteurs. Il faudrait donc trouver un moyen de partager les données de contrôle et les données des instances de processus, et explorer les différentes possibilités quant au choix du langage de coordination des systèmes d'exécution de workflow. Une autre amélioration serait d'inclure, dans la liste des applications de workflow, des web services, transformant ainsi une définition de processus en un enchaînement de fonctions serveur. Une dernière piste pourrait être de prévoir un outil graphique permettant de visualiser et modéliser des processus (en utilisant les langages XPDL et BPMN par exemple) prévus pour s'exécuter sur plusieurs moteurs de workflow.

Annexe A

Définition de processus en XPDL

Voici le code XML de la définition de processus XPDL construite au chapitre 5.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package Id ="0" Name="sample workflow process">
  <TypeDeclarations>
    <TypeDeclaration Id="TYPE_detail_d_une_commande"
      Name="detail_d_une_commande">
      <ExternalReference
        location="http://localhost/detail_d_une_commande.xsd" />
    </TypeDeclaration>
    <TypeDeclaration Id="TYPE_rapport_sur_le_stock"
      Name="rapport_sur_le_stock">
      <ExternalReference
        location="http://localhost/rapport_sur_le_stock.xsd" />
    </TypeDeclaration>
    <TypeDeclaration Id="TYPE_information_sur_le_reapprovisionnement"
      Name="information_sur_le_reapprovisionnement">
      <ExternalReference
        location="http://localhost/information_sur_le_reapprovisionnement.xsd" />
    </TypeDeclaration>
  </TypeDeclarations>
  <WorkflowProcesses>
    <WorkflowProcess Id="1" Name="commande_de_produits" AccessLevel="PUBLIC">
      <DataFields>
        <DataField Id="detail_de_la_commande">
          <DataType>
            <DeclaredType Id="TYPE_detail_d_une_commande" />
          </DataType>
        </DataField>
        <DataField Id="numero_de_reference_de_la_commande">
          <DataType>
            <BasicType Type="INTEGER" />
          </DataType>
        </DataField>
      </DataFields>
    </WorkflowProcess>
  </WorkflowProcesses>
</Package>
```

```

</DataField>
<DataField Id="rapport_sur_le_stock">
  <DataType>
    <DeclaredType Id="TYPE_rapport_sur_le_stock" />
  </DataType>
</DataField>
<DataField Id="information_sur_le_reapprovisionnement">
  <DataType>
    <DeclaredType
      Id="TYPE_information_sur_le_reapprovisionnement" />
    </DataType>
  </DataField>
</DataFields>
<Participants>
  <Participant Id="Client">
    <ParticipantClass Type="HUMAN"/>
    <Description>un utilisateur particulier qui commande des
      produits</Description>
  </Participant>
  <Participant Id="Traitement_automatique">
    <ParticipantClass Type="SYSTEM"/>
    <Description>des applications liées aux activités sans
      intervention humaine</Description>
  </Participant>
  <Participant Id="Employe">
    <ParticipantClass Type="ROLE"/>
    <Description>une personne de l'organisation ayant les
      qualifications requises</Description>
  </Participant>
</Participants>
<Applications>
  <Application Id="application_de_commande_des_articles">
    <FormalParameters>
      <FormalParameter Id="FP_detail_de_la_commande"
        Index="1" Mode="OUT">
        <DataType>
          <DeclaredType Id="TYPE_detail_d_une_commande" />
        </DataType>
      </FormalParameter>
    </FormalParameters>
  </Application>
  <Application Id="application_d_enregistrement_de_la_commande">
    <FormalParameters>
      <FormalParameter Id="FP_detail_de_la_commande"
        Index="1" Mode="IN">
        <DataType>
          <DeclaredType Id="TYPE_detail_d_une_commande" />
        </DataType>
      </FormalParameter>
    </FormalParameters>
  </Application>
</Applications>

```

```

        </DataType>
    </FormalParameter>
    <FormalParameter Id="FP_numero_de_reference_de_la_commande"
        Index="2" Mode="OUT">
        <DataType>
            <BasicType Type="INTEGER" />
        </DataType>
    </FormalParameter>
</FormalParameters>
</Application>
<Application Id="application_de_verification_du_stock">
    <FormalParameters>
        <FormalParameter Id="FP_numero_de_reference_de_la_commande"
            Index="1" Mode="IN">
            <DataType>
                <BasicType Type="INTEGER" />
            </DataType>
        </FormalParameter>
        <FormalParameter Id="FP_rapport_sur_le_stock"
            Index="2" Mode="OUT">
            <DataType>
                <DeclaredType Id="TYPE_rapport_sur_le_stock" />
            </DataType>
        </FormalParameter>
    </FormalParameters>
</Application>
<Application
    Id="application_d_information_au_client_sur_le_manque_de_stock">
    <FormalParameters>
        <FormalParameter Id="FP_rapport_sur_le_stock" Index="1" Mode="IN">
            <DataType>
                <DeclaredType Id="TYPE_rapport_sur_le_stock" />
            </DataType>
        </FormalParameter>
        <FormalParameter Id="FP_numero_de_reference_de_la_commande"
            Index="2" Mode="IN">
            <DataType>
                <BasicType Type="INTEGER" />
            </DataType>
        </FormalParameter>
    </FormalParameters>
</Application>
<Application Id="application_de_reapprovisionnement_de_stock">
    <FormalParameters>
        <FormalParameter Id="FP_rapport_sur_le_stock" Index="1" Mode="IN">
            <DataType>
                <DeclaredType Id="TYPE_rapport_sur_le_stock" />

```

```

        </DataType>
    </FormalParameter>
    <FormalParameter
        Id="FP_information_sur_le_reapprovisionnement"
        Index="2" Mode="OUT">
        <DataType>
            <DeclaredType
                Id="TYPE_informations_sur_le_reapprovisionnement" />
        </DataType>
    </FormalParameter>
</FormalParameters>
</Application>
<Application Id="application_d_information_au_client_
    sur_le_reapprovisionnement_du_stock">
    <FormalParameters>
        <FormalParameter Id="FP_numero_de_reference_de_la_commande"
            Index="1" Mode="IN">
            <DataType>
                <BasicType Type="INTEGER" />
            </DataType>
        </FormalParameter>
        <FormalParameter Id="FP_information_sur_le_reapprovisionnement"
            Index="2" Mode="IN">
            <DataType>
                <DeclaredType
                    Id="TYPE_information_sur_le_reapprovisionnement" />
            </DataType>
        </FormalParameter>
    </FormalParameters>
</Application>
<Application Id="application_d_envoi_de_la_commande">
    <FormalParameters>
        <FormalParameter Id="FP_numero_de_reference_de_la_commande"
            Index="1" Mode="IN">
            <DataType>
                <BasicType Type="INTEGER" />
            </DataType>
        </FormalParameter>
    </FormalParameters>
</Application>
<Application Id="application_d_information_au_client_de_
    l_envoi_de_la_commande">
    <FormalParameters>
        <FormalParameter Id="FP_numero_de_reference_de_la_commande"
            Index="1" Mode="IN">
            <DataType>
                <BasicType Type="INTEGER" />

```

```

        </DataType>
    </FormalParameter>
</FormalParameters>
</Application>
</Applications>
<Activities>
    <Activity Id="1" Name="commande_des_articles">
        <Implementation>
            <Tool Id="application_de_commande_des_articles"
                Type="APPLICATION">
                <ActualParameters>
                    <ActualParameter>detail_de_la_commande</ActualParameter>
                </ActualParameters>
            </Tool>
        </Implementation>
        <Performer>Client</Performer>
    </Activity>
    <Activity Id="2" Name="enregistre_la_commande">
        <Implementation>
            <Tool Id="application_d_enregistrement_de_la_commande"
                Type="APPLICATION">
                <ActualParameters>
                    <ActualParameter>detail_de_la_commande</ActualParameter>
                    <ActualParameter>
                        numero_de_reference_de_la_commande</ActualParameter>
                </ActualParameters>
            </Tool>
        </Implementation>
        <Performer>Traitement_automatique</Performer>
    </Activity>
    <Activity Id="3" Name="verifie_le_stock">
        <Implementation>
            <Tool Id="application_de_verification_du_stock"
                Type="APPLICATION">
                <ActualParameters>
                    <ActualParameter>
                        numero_de_reference_de_la_commande</ActualParameter>
                    <ActualParameter>rapport_sur_le_stock</ActualParameter>
                </ActualParameters>
            </Tool>
        </Implementation>
        <Performer>Traitement_automatique</Performer>
    </Activity>
    <Activity Id="4" Name="informe_le_client_que_le_stock_est_insuffisant">
        <Implementation>
            <Tool Id="application_d_information_au_client_sur_le_manque_de_stock"
                Type="APPLICATION">

```

```

        <ActualParameters>
            <ActualParameter>rapport_sur_le_stock</ActualParameter>
            <ActualParameter>
                numero_de_reference_de_la_commande</ActualParameter>
        </ActualParameters>
    </Tool>
</Implementation>
<Performer>Traitement_automatique</Performer>
</Activity>
<Activity Id="5" Name="reapprovisionne_le_stock">
    <Implementation>
        <Tool Id="application_de_reapprovisionnement_de_stock"
            Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>rapport_sur_le_stock</ActualParameter>
                <ActualParameter>
                    information_sur_le_reapprovisionnement</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <Performer>Employe</Performer>
</Activity>
<Activity Id="6"
    Name="informe_le_client_que_le_stock_est_reapprovisionne">
    <Implementation>
        <Tool Id="application_d_information_au_client_sur_le
            _reapprovisionnement_du_stock" Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>
                    numero_de_reference_de_la_commande</ActualParameter>
                <ActualParameter>
                    information_sur_le_reapprovisionnement</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <Performer>Traitement_automatique</Performer>
</Activity>
<Activity Id="7" Name="envoie_la_commande">
    <Implementation>
        <Tool Id="application_d_envoi_de_la_commande"
            Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>
                    numero_de_reference_de_la_commande</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>

```

```

    <Performer>Client</Performer>
  </Activity>
  <Activity Id="8" Name="informe_le_client_de_l_envoi_de_la_commande">
    <Implementation>
      <Tool
        Id="application_d_information_au_client_de_l_envoi_de_la_commande"
        Type="APPLICATION">
          <ActualParameters>
            <ActualParameter>
              numero_de_reference_de_la_commande</ActualParameter>
            </ActualParameters>
          </Tool>
        </Implementation>
      <Performer>Traitement_automatique</Performer>
    </Activity>
    <Activity Id="9" Name="decision_split_xor">
      <Route/>
      <TransitionRestrictions>
        <TransitionRestriction>
          <Split Type="XOR">
            <TransitionRefs>
              <TransitionRef Id="5" />
              <TransitionRef Id="9" />
            </TransitionRefs>
          </Split>
        </TransitionRestriction>
      </TransitionRestrictions>
    </Activity>
    <Activity Id="10" Name="split_and">
      <Route/>
      <TransitionRestrictions>
        <TransitionRestriction>
          <Split Type="AND">
            <TransitionRefs>
              <TransitionRef Id="10" />
              <TransitionRef Id="11" />
            </TransitionRefs>
          </Split>
        </TransitionRestriction>
      </TransitionRestrictions>
    </Activity>
    <Activity Id="11" Name="join_and">
      <Route/>
      <TransitionRestrictions>
        <TransitionRestriction>
          <Join Type="AND" />
        </TransitionRestriction>
      </TransitionRestrictions>
    </Activity>
  </Activity>

```



```

        </TransitionRestrictions>
    </Activity>
    <Activity Id="12" Name="join_xor">
        <Route/>
        <TransitionRestrictions>
            <TransitionRestriction>
                <Join Type="XOR" />
            </TransitionRestriction>
        </TransitionRestrictions>
    </Activity>
    <Activity Id="13" Name="debut">
        <Route/>
    </Activity>
    <Activity Id="14" Name="fin">
        <Route/>
    </Activity>
</Activities>
<Transitions>
    <Transition Id="1" From="13" To="1"/>
    <Transition Id="2" From="1" To="2"/>
    <Transition Id="3" From="2" To="3"/>
    <Transition Id="4" From="3" To="9"/>
    <Transition Id="5" From="9" To="12">
        <Condition>rapport_sur_le_stock.ok == true</Condition>
    </Transition>
    <Transition Id="6" From="12" To="7"/>
    <Transition Id="7" From="7" To="8"/>
    <Transition Id="8" From="8" To="14"/>
    <Transition Id="9" From="9" To="10">
        <Condition>rapport_sur_le_stock.ok == false</Condition>
    </Transition>
    <Transition Id="10" From="10" To="4"/>
    <Transition Id="11" From="10" To="13"/>
    <Transition Id="12" From="4" To="11"/>
    <Transition Id="13" From="5" To="11"/>
    <Transition Id="14" From="11" To="6"/>
    <Transition Id="15" From="6" To="12"/>
</Transitions>
</WorkflowProcess>
</WorkflowProcesses>
</Package>

```

Bibliographie

- [1] Free online dictionary of computing. <http://wombat.doc.ic.ac.uk/foldoc>.
- [2] Merriam webster online. <http://www.m-w.com>.
- [3] RosettaNet. <http://www.wfmc.org>.
- [4] WfMC. <http://www.wfmc.org>.
- [5] Sinan Si Alhir. *Introduction à UML*. Modélisation de projets. Editions O'Reilly, Paris, 2004. ISBN 0-9703509-4-5.
- [6] Tom Bayens. The state of the workflow. <http://www.jboss.org/products/jbpm/stateofworkflow>.
- [7] David Hollingsworth. *Workflow Management Coalition The Workflow Reference Model*. Workflow Management Coalition, 2 Crown Walk Winchester Hampshire, UK SO22 5XE, January 1995. issue 1.1.
- [8] Pascal Roques. *UML. Les cahiers du programmeur*. Editions Eyrolles, 61, Bld Saint-Germain 75240 Paris Cedex 05, 2002. ISBN 0-9703509-4-5.
- [9] Alec Sharp and Patrick McDermott. *Workflow Modeling Tools for Process Improvement and Application Development*, chapter 2. Artech House, Inc, 2001.
- [10] Alec Sharp and Patrick McDermott. *Workflow Modeling Tools for Process Improvement and Application Development*, pages 3–4. Artech House, Inc, 2001.
- [11] Alec Sharp and Patrick McDermott. *Workflow Modeling Tools for Process Improvement and Application Development*, chapter 4. Artech House, Inc, 2001.
- [12] Dr. Stephen A. White. Process modeling notations and workflow patterns. In Layna Fischer, editor, *Workflow handbook 2004*, pages 265–293. Future Strategies Inc., Book Division, Lighthouse point, Florida, 2004. ISBN 0-9703509-6-1.
- [13] Stephen A. White. XPDL and BPMN. In Layna Fischer, editor, *Workflow handbook 2003*, pages 221–238. Future Strategies Inc., Book Division, Lighthouse point, Florida, 2003. ISBN 0-9703508-4-5.
- [14] Workflow Management Coalition. *Workflow Management Application Programming Interface (Interface 2 & 3) Specification*, July 1998. version 2.0.
- [15] Workflow Management Coalition, 2 Crown Walk Winchester Hampshire, UK SO23 8BB. *Workflow Management Coalition Terminology & Glossary*, February 1999. version 3.0.

- [16] Workflow Management Coalition, 2436, N. Federal Highway #374, Lighthouse Point, Fl 33064. *Workflow Process Definition Interface – XML Process Definition Language*, October 2002. version 1.0.